University of Alberta

A Disease Classifier for Metabolic Profiles Based on Metabolic Pathway Knowledge

by

Thomas Eastman

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

Spring 2010
Edmonton, Alberta

Examining Committee

Russell Greiner, Computing Science

Vickie Baracos, Oncology

Dale Schuurmans, Computing Science

Abstract

This thesis presents Pathway Informed Analysis (PIA), a classification method for predicting disease states (diagnosis) from metabolic profile measurements that incorporates biological knowledge in the form of metabolic pathways. A metabolic pathway describes a set of chemical reactions that perform a specific biological function. A significant amount of biological knowledge produced by efforts to identify and understand these pathways is formalized in readily accessible databases such as the Kyoto Encyclopedia of Genes and Genomes. PIA uses metabolic pathways to identify relationships among the metabolite concentrations that are measured by a metabolic profile. Specifically, PIA assumes that the class-conditional metabolite concentrations (diseased vs. healthy, respectively) follow multivariate normal distributions. It further assumes that conditional independence statements about these distributions derived from the pathways relate the concentrations of the metabolites to each other. The two assumptions allow for a natural representation of the class-conditional distributions using a type of probabilistic graphical model called a Gaussian Markov Random Field. PIA efficiently estimates the parameters defining these distributions from example patients to produce a classifier. It classifies an undiagnosed patient by evaluating both models to determine the most probable class given their metabolic profile.

We apply PIA to a data set of cancer patients to diagnose those with a muscle wasting disease called cachexia. Standard machine learning algorithms such as Naïve Bayes, Tree-augmented Naïve Bayes, Support Vector Machines and C4.5 are used to evaluate the performance of PIA. The overall classification accuracy of PIA is better than these algorithms on this data set but the difference is not statistically significant. We also apply PIA to several other classification tasks. Some involve predicting various manipulations of the metabolic processes performed in experiments with worms. Other tasks are to classify pigs according to properties of their dietary intake. The accuracy of PIA at these tasks is not significantly better than the standard algorithms.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This thesis considers the problem of diagnosing a disease from the metabolic profile of a patient. A metabolic profile quantifies the concentrations of various chemical compounds, called metabolites, that are present in the body. Metabolites are small molecules produced by breaking down larger molecules and consumed to build other molecules that the body needs through processes called catabolism and anabolism respectively [37]. Together these processes compose the metabolism of an organism. Here, we focus on a disease called cachexia that occurs when the body catabolizes its own muscle tissue. Intuitively, a metabolic profile will convey some information relevant to the cachexic state of a patient. Catabolism of muscle tissue produces various metabolites that we expect to increase in concentration in affected patients. Currently, no simple diagnostic rule based on metabolite concentrations exists for cachexia however. Positively diagnosing cachexia currently requires an expert to examine diagnostic images of a patient, such as those from computerized tomography (CT) scans, at two different time points to identify a loss of muscle mass [35]. The goal is to develop an automated disease diagnosis tool for a disease such as cachexia. Our approach is unique because it combines knowledge of human metabolism with known examples of patients that are positive and negative for that disease. These examples are in the form of a single metabolic profile for each patient. The underlying ideas are general enough that they are potentially applicable to diagnosing any disease affecting the metabolism of a patient.

This type of problem fits naturally into the framework of supervised machine learning. In this setting, the goal is to produce a model that can classify objects based on some description of them. For the problem of disease diagnosis, patients are described by their metabolite concentrations and they are from one of two categories: they have the disease or they do not. The model is typically derived from examples of each of the relevant categories. An algorithm that produces a classifier from examples previously classified by a domain expert is called a learning algorithm. Research in supervised machine learning has produced numerous learning algorithms to address these classification problems. Implementations of these algorithms that can produce diagnosis tools based on metabolic profile data are readily available. Performance of these tools is measured by their accuracy at diagnosing new patients. Our experience with some standard machine learning methods shows that they incorrectly diagnose many patients. This thesis attempts to improve on these results by taking advantage of an easily accessible and intuitively useful form of expert biological knowledge.

We consider knowledge of metabolic pathways to improve classification accuracy for metabolic profile data. A metabolic pathway is a set of chemical reactions that perform a specific function [37]. These reactions are linked to each other in sequences where the products of one reaction are consumed in the next. Human metabolism involves many such pathways that have been elucidated over decades of biological research. Databases such as

the Kyoto Encyclopedia of Genes and Genomes (KEGG) formalize the description of these pathways for a variety of organisms. In KEGG, pathways relate metabolites via the reaction equations that compose the known metabolic pathways for each organism [28]. We expect that knowing how measured metabolites participate in chemical reactions with each other in the body will improve the performance of classifiers based on metabolic profile data. This thesis considers an interpretation of the metabolic pathways as probabilistic independence statements about the distributions of metabolite concentration values. Graph structures are a natural way to represent metabolic pathways leading to consideration of classifiers based on probabilistic graphical models.

We develop a diagnosis tool around these ideas called Pathway Informed Analysis (PIA). The system effectively incorporates expert biological knowledge of metabolic pathways and it efficiently learns parameters from a data set of example metabolic profiles. Evaluating the model learned by PIA to diagnose a new patient is also efficient. PIA results in better accuracy than standard classification methods on metabolic profile data from cancer patients with and without cachexia. The difference between PIA and the other methods is not statistically significant however. This suggests that metabolic pathways from KEGG may provide some benefit in performing classification based on metabolic profile data.

1.1 Motivation

One reason that classifiers based on metabolic profile data are easy to implement is because a usable biological sample is relatively easy to obtain. Our cachexia data set contains metabolic profiles derived from urine samples. Urine is a biological fluid that is minimally invasive to acquire and that is routinely collected from patients. Intuitively, urinary metabolic profiles should convey some signal indicating the cachexic state of a patient. As suggested, we expect that catabolism of muscle tissue will increase the concentrations of certain metabolites in the body. Some of these metabolites are waste products that are removed from the body via urine [18]. For reasons discussed below, retrieving this signal from urinary metabolic profiles is a challenging task. It is possible to obtain metabolic profiles from other samples such as blood or tissue. Muscle tissue samples are desirable for diagnosing cachexia because this is where the disease takes place but these samples require removing a portion of muscle tissue from a patient. This is a much more invasive procedure than collecting a urine sample. Our goal is to take advantage of the more readily available form of data by using biological knowledge to overcome its limitations.

Extensive knowledge of human metabolism in the form of metabolic pathways is available in databases such as KEGG. This knowledge is provided in a format that is easily parsed to retrieve information about the pathways including the reactions and metabolites that compose them. Previous research has considered graph representations of metabolic pathways and has produced a number of possibilities. Graph structures are the basis for graphical models that provide a way to represent probabilistic independence statements about a set of variables and to compactly represent probability distributions over them [30]. The compactness of the representation is greater the smaller the number of nodes and edges in the underlying graph. Because metabolic pathway graphs are sparse, PIA should benefit from a reduction in the number of parameters it must learn to represent distributions over metabolite concentrations. Fewer parameters reduces the likelihood of overfitting, a phenomenon where a learned model captures too much of the noise in a data set and fails to accurately classify future examples [22]. We hypothesize that PIA will learn more accurate parameter values that will result in better classification performance when compared to other learning algorithms.

Computational diagnosis of cachexia from metabolic profiles is potentially more efficient than current approaches. These approaches are based on diagnostic images of a patient

acquired at two different times that an expert analyzes to determine if the patient has lost muscle mass or not [35]. The time between these images represents a delay that contributes to the poor prognosis of cachexia [36]. Naturally, an early positive diagnosis of a disease means that treatment for it can start sooner. An automated diagnosis tool based on single time point urinary metabolic profiles would reduce the time needed to make a positive diagnosis of cachexia. This is an important achievement because often cachexia is associated with cancer. For a cachexic patient, certain cancer treatments are harmful [33, 34], thus efficiently diagnosing cachexia in patients with cancer is important for their cancer treatment as well. In addition, an automated diagnosis system may reduce the amount of intervention required from trained human experts, freeing them for other tasks. Applying machine learning techniques to make diagnoses from urinary metabolic profiles poses significant challenges however.

1.2 Challenges

Cachexia diagnosis from urinary metabolic profile data is challenging for several reasons. The relevant biological state of this disease occurs within the muscle tissue of an affected patient. A urine sample only indirectly reflects this state because all bodily tissues produce waste products that are excreted via the urine [18]. Thus, the waste products that we expect to increase in concentration when muscle is catabolized are diluted with the waste products from other biological processes. Relevant concentration changes indicative of cachexia are more difficult to detect as a result. As suggested, muscle tissue is a possible source for metabolic profiles but acquiring the samples is more invasive of the patient. PIA seeks to address the challenges of working with more easily acquired urinary metabolic profile data.

Another problem specific to urinary metabolic profiles is related to kidney function. The kidney is an organ that actively influences the metabolites that appear in the urine. Through a process called re-uptake, the kidneys selectively draw certain useful metabolites, some produced by muscle catabolism, from the urine back into the body before the urine is excreted [18]. The efficiency of this process is limited however, thus some useful metabolites will appear in a urine sample. The selectivity of the process interferes with detection of concentration differences between positive and negative patients. Urinary metabolite concentrations for waste metabolites and useful metabolites reflect their actual concentrations in the body differently. The metabolic pathway knowledge that PIA uses may suggest that the concentrations of metabolites on the same catabolic pathway are all elevated in cachexic patients but kidney function makes this less apparent in the data.

Working with urinary metabolic profile data from patients poses challenges specific to human subjects. In addition to disease, many factors influence urinary metabolite concentrations such as age, sex, diet, drugs, etc. Digestion of certain food produces byproducts similar to those produced by muscle catabolism for example. These factors are not controlled in our cachexia data set. Naturally, it is difficult to control these variables in a population of human patients. All of these factors contribute noise with respect to disease state in the measured metabolite concentrations because they are mostly irrelevant to the correct diagnosis. Factors such as these complicate the process of learning effective disease classifiers from a human metabolic profile data set.

The cachexia data set that we consider here is limited in some other ways as well. It contains 73 patients, each with a single metabolic profile derived from a spot urine sample. These are samples that are taken at a single time point, thus the metabolic profiles exhibit higher variability than those based on samples taken over a period of time such as 24 hour urine samples. Each metabolic profile in the data set contains 63 measured metabolite concentrations, a number that is large relative to the number of patients. The dimensionality of the metabolic profile measurements is problematic for another reason as well. The 63

measured metabolites represent only a small fraction of the approximately 1,500 metabolites represented in the full set of human metabolic pathways in KEGG. Constructing graphical models that directly represent these pathways is not feasible because they would contain far too many unobserved (latent) variables to handle effectively. Although we cannot change the properties of the data set, we will consider various approaches to eliminating latent variables from the models that PIA uses. A related problem is that only some of the 63 measured metabolites are represented in the set of KEGG metabolic pathways. Those that are not represented require special handling to avoid losing this portion of the data set.

1.3 Assumptions

PIA makes several assumptions that help it address the challenges outlined above. These assumptions are necessary because most machine learning algorithms were not designed to incorporate the type of expert knowledge that we are considering. They are made in order to incorporate metabolic pathways into PIA in a way that allows for efficient parameter learning and model evaluation to diagnose new patients. Supervised machine learning algorithms generally assume that a data set of classified examples is provided from which to derive a classification model. Metabolic pathways constitute a different form of input. PIA assumes that graph structures are an appropriate representation for metabolic pathway knowledge because graphs are a natural representation for computational analysis. The result is a classification problem involving noisy measurements of multiple variables and a graph structure relating the variables. Probabilistic graphical models are an appropriate choice to address this type of problem.

Using KEGG metabolic pathways to derive a probabilistic graphical model structure rests on the assumptions that the KEGG pathways are correct and complete. Correctness means that there are no errors in the pathways that KEGG contains. Completeness implies that KEGG contains all known pathways, including all of their constituent reactions. The latter assumption is false because KEGG is missing some of the metabolites in our data set and known reactions involving them. We take steps to correct these problems (described in Appendix A) because of their relevance to the metabolic mechanisms of cachexia. In general, we do not know how faithful KEGG is in representing human metabolism but there are reasons to believe that the two assumptions are mostly correct. First, metabolism is well studied and understood at a high level of abstraction. Detailed parameters governing metabolic reactions such as reaction rates are not well understood for all reactions but the metabolites and their participation in metabolic reactions are [37]. KEGG originated in 1995 and it is one of several competing metabolic pathway databases [26]. The literature does not suggest that KEGG is inferior to any of these at meeting the requirements of PIA. It is possible that none of these databases contains an adequate representation of metabolic pathways to support the probabilistic interpretation we apply to them however.

KEGG metabolic pathways and probabilistic graphical models both have semantics that PIA must unify in an appropriate manner. The graph structure of a probabilistic graphical model encodes conditional independence statements about its associated variables [30]. Thus, PIA assumes a probabilistic interpretation of metabolic pathways to model distributions of metabolite concentrations in a meaningful way. Metabolic pathways in KEGG are described as sets of chemical reaction equations. From these equations, PIA can determine, for any metabolite v, the set of metabolites $N(v)$ that are involved with v in some reaction. Suppose that the concentration value of each metabolite in the set $N(v)$ is measured. PIA assumes that the concentration of v is independent of every other metabolite in the body given these measurements. This makes sense because, according to KEGG, there is no way to directly influence the production of consumption of metabolite v that does not involve the metabolites in $N(v)$. Conditional independence statements about metabolite concentra-

tions of this form lead to a simple method for constructing a graph from a set of metabolic pathways as described in Section 3.2.

PIA makes the additional assumption that all metabolic reactions can operate in both directions. Although true for many reactions, some only convert matter in one direction [37]. For these, the substrate metabolites and the products that are derived from them are fixed. PIA assumes that the sets of substrates and products in a reaction are always interchangeable. This assumption is convenient in the context of probabilistic graphical models because it is difficult to accommodate graph structures with directed cycles. We are not aware of a sensible graph representation of metabolic pathways that avoids cycles. To address this problem, PIA will consider only undirected graph structures because graphical models based on them effectively incorporate cyclic graphs. The assumption is also mild from a biological perspective. Most metabolic reactions depend on an enzyme to catalyze them. Because of the way these enzymes work, concentrations of product metabolites can affect the concentrations of the metabolites used to produce them. Thus, concentration changes can propagate backward through irreversible reactions even if actual matter does not [45].

Another assumption is related to the steady state of metabolic reactions. When a reaction is in steady state, the rates that it consumes substrate metabolites and produces product metabolites are equal [37]. Because our data set contains a single time point metabolic profile for each patient, PIA implicitly assumes that the entire metabolic network is in steady state with respect to the cachexic state of the patient. This means that when the disease occurs, it exerts some change on the consumption and production rates of each metabolic reaction. After the initial changes, the disease does not continue to effect changes in these rates. From the perspective of the classifier, factors other than the disease of interest that influence them are summarized in the metabolic profile measurements as noise. Thus, PIA does not assume that the reactions themselves are in steady state. It only assumes that the disease of interest is not characterized by changing rates of consumption and production of metabolites. Classifiers for a disease with this characteristic would benefit from metabolic profile measurements taken at multiple time points.

PIA further assumes that the class-conditional metabolite concentrations follow multivariate normal distributions to ensure efficient learning and inference. This means that the metabolic profiles of diseased patients are described by a multivariate normal distribution and those of healthy patients follow a different multivariate normal distribution. The assumption is not entirely unrealistic because metabolic profile measurements are continuous values and the univariate distributions of these values are approximately unimodal. It allows PIA to take advantage of a type of undirected graphical model called a Gaussian Markov Random Field (GMRF). Parameter learning and classification of undiagnosed patients are both computationally efficient with these models.

1.4 Classification Technique

To more formally outline the classification technique of PIA, this section introduces notation used throughout this thesis. The metabolic profile of a patient r is denoted by a vector $x_r = (x_{r1}, x_{r2}, \ldots, x_{rp})^T$ that contains concentration values for p measured metabolites. Specifically, the variable $x_{ri} \in \mathbb{R}$ represents the concentration value of metabolite i for the patient r. A generic metabolic profile, without reference to a specific patient, is denoted by $x = (x_1, x_2, \ldots, x_p)^T$. A patient r has an associated class $c_r \in \{+, -\}$ that indicates their disease state. The class $c_r = +$ indicates that patient r is positive for the disease whereas $c_r = -$ indicates that r is negative for it. PIA assumes that it is provided a data set $\mathcal{D}$ containing some number of example patients $n = n^+ + n^-$ where n^+ are positive and n^- are negative. It further assumes that the correct class c_r is given for each patient in

the data set. PIA learns a model from these examples that accurately predicts the class of undiagnosed patients based only on their metabolic profiles.

As noted above, PIA assumes that a metabolic profile x_r follows one of two multivariate normal distributions. Metabolic profiles of positive patients are distributed according to one and those of negative patients according to the other. These distributions are called class-conditional distributions because they depend on the class of the patient. A multivariate normal distribution is usually described by two parameters: a mean vector μ and a covariance matrix Σ. The probability density function is

$$P(x_r \mid \mu, \Sigma) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x_r - \mu)^T \Sigma^{-1}(x_r - \mu)\right)$$

for the given parameters [30]. The development of PIA is oriented around the inverse of the covariance matrix Σ^{-1}. For convenience, we denote this matrix, also called the precision matrix [30], by $K = \Sigma^{-1}$. Note that we can parameterize the density function with K instead of Σ by replacing Σ with K^{-1}. Since PIA is concerned with class-conditional distributions, we must distinguish between the class-conditional parameters. We use μ^+ and K^+ (or Σ^+) to denote the parameters of the distribution for the positive patients and μ^- and K^- (or Σ^-) for the negative patient distribution. The learning component of PIA consists of estimating these parameter values from the metabolic profiles of example patients given a graph structure describing conditional independence statements about the distributions. Various algorithms employed by PIA for this task are discussed in Chapter 4.

After estimating the class-conditional parameter values, PIA classifies a new patient r with metabolic profile x_r as follows. PIA assigns the patient class c_r as positive for the disease if

$$P(c_r = + \mid x_r) > P(c_r = - \mid x_r)$$

and as negative otherwise. Unfortunately, the posterior probability values $P(c_r = + \mid x_r)$ and $P(c_r = - \mid x_r)$ are not directly computable from the learned parameters. PIA can use these learned parameters to compute the class-conditional densities $P(x_r \mid c_r = +)$ and $P(x_r \mid c_r = -)$ however. The posterior probability values are then computed indirectly using Bayes' rule [30]. For example

$$P(c_r = + \mid x_r) = \frac{P(x_r \mid c_r = +)P(c_r = +)}{P(x_r)}$$

and similarly for $P(c_r = - \mid x_r)$. The classification rule is now equivalent to

$$P(x_r \mid c_r = +)P(c_r = +) > P(x_r \mid c_r = -)P(c_r = -)$$

where the denominator terms $P(x_r)$ are dropped because they cancel each other. Computing the values in this rule requires the class prior probabilities $P(c_r = +)$ and $P(c_r = -)$ but these are easily computed by counting patients in the data set, for example $P(c_r = +) = \frac{n^+}{n}$. Although this method can be employed as described with no prior biological knowledge, PIA is based on the hypothesis that incorporating this knowledge will improve classification accuracy.

PIA takes advantage of metabolic pathway knowledge through its use of GMRF models. A GMRF model represents a multivariate normal distribution by explicitly incorporating conditional independence statements about the variables in the distribution [30]. These conditional independence statements are represented as an undirected graph where each node corresponds to a variable and each edge encodes a dependence between two variables. In particular, the value of a variable is independent of the other variables given the values of its neighbors in the graph [30]. Section 3.2 discusses various graph structure representations of metabolic pathways. We will show that one approach yields conditional independence statements with an intuitively appealing interpretation. Under this interpretation,

the concentration of a metabolite is independent of all others given the concentrations of the metabolites participating in a common reaction with it. Knowing conditional independence statements such as these has immediate relevance for the way that PIA learns parameters.

The undirected graph structure of a GMRF describes a pattern of zero-valued elements in the precision matrix of the corresponding distribution. The precision matrix element corresponding to two variables is zero if the nodes corresponding to those variables are not joined by an edge [30]. Because most metabolites participate in only a few reactions as observed in KEGG, we expect that a graph of all metabolic pathways will be sparse. In other words, the graph will contain few edges resulting in many zero-valued elements. Knowing that an element of the precision matrix is zero means that PIA does not have to estimate it from the data set. This is important because it reduces the possibility that the learning algorithm will overfit the data causing poor performance on undiagnosed patients [22]. A GMRF structure derived from a set of metabolic pathways implies a set of constraints on the precision matrix of the distribution. Fortunately, efficient algorithms exist that can perform parameter estimation for multivariate normal distributions subject to these constraints. PIA employs these to learn the parameters of the distributions corresponding to positive and negative patients.

1.5 Contribution

This thesis develops a tool called PIA that diagnoses a single disease from the metabolic profile of a patient. It learns to distinguish patients with the disease from those without by analyzing example metabolic profiles of patients from both categories. PIA incorporates biological knowledge through a novel probabilistic interpretation of metabolic pathways as conditional independence statements about the joint distribution of metabolite concentrations. This interpretation allows PIA to take advantage of existing learning algorithms based on probabilistic graphical models. The contribution of this thesis is a description of how to make a disease classifier from existing components. In the chapters that follow, we describe the ideas that PIA employs to address the challenges outlined above. The resulting system is general enough that it is applicable to any diagnosis problem based on metabolic profile measurements. Additional ideas for extending and possibly improving PIA are suggested as well. PIA is evaluated empirically using a data set of urinary metabolic profiles from real cancer patients with and without cachexia.

Several well-established machine learning algorithms are used to provide a benchmark for evaluating PIA. The algorithms include Naïve Bayes, Tree-augmented Naïve Bayes, Support Vector Machines and C4.5 decision trees. We use cross validation to determine the overall accuracy of each approach on all patients in the data set. The results show that PIA obtains higher classification accuracy than any other method. The difference between PIA and the other top performing algorithms is not statistically significant however. Thus it is not clear from our work that PIA is a superior method for diagnosing patients from their metabolic profiles. PIA is based on several assumptions discussed above that may limit the best performance that it can achieve. To our knowledge, this is the first attempt to incorporate metabolic pathway knowledge and metabolic profile data into a supervised learning algorithm.

Chapter 2

Background

The development of the PIA system depends on fundamental concepts from biology and machine learning. PIA incorporates knowledge about certain chemical compounds involved in human metabolism and their interactions with each other. Here, we discuss this knowledge including how it is organized in a biological database called the Kyoto Encyclopedia of Genes and Genomes (KEGG). Ideas from machine learning are important to the development of PIA as they provide the foundation of the system as a disease classifier. These ideas are also important because we will evaluate the performance of PIA by comparing it to existing machine learning methods.

2.1 Metabolic Profiles

Metabolic profiles quantify the concentration of various chemical compounds in a biological sample. These compounds, called metabolites, are small molecules that are consumed to make the large molecules that the body needs and are produced by breaking down those that are not needed [37]. Intuitively, a metabolic profile can help diagnose cachexia because the disease is an abnormal breakdown of muscle tissue that produces some of the measured metabolites. A metabolic profile can be derived from different samples including urine, blood and tissue. From the perspective of a classifier, the sample source most relevant to the disease of interest is ideal but, for human patients, a sample that is minimally invasive to collect is desirable. The ideal sample source for cachexia is muscle tissue because this is where the disease occurs. Here, we focus on urine samples to quantify metabolite concentrations as these are easier to collect.

The quantification of metabolites is performed via a process called nuclear magnetic resonance (NMR) spectroscopy. Jacobsen describes the process as follows [24]. The NMR spectroscopy technique focuses on one particular atom. For example, our cachexia data set focuses on the ^{1}H atom, an isotope of hydrogen, thus it is called ^{1}H-NMR spectroscopy. When put into a magnetic field, nuclei of this atom resonate at a certain known frequency. Within a particular compound, various atoms are joined together in an arrangement unique to that compound. Atoms joined to ^{1}H atoms influence its resonant frequency, shifting it slightly depending on the specific arrangement. These shifts are also known, thus each compound produces a unique signature in the spectrum of resonant frequencies. Experts can identify compounds in the spectrum based on these signatures. An example spectrum produced from a urine sample of a patient in our cachexia data set is given in Figure 2.1. Quantifying metabolite concentrations involves identifying the patterns of peaks in a spectrum that correspond to different compounds. The intensity of the signal at a given frequency is proportional to the amount of ^{1}H resonating at that frequency. Experts can determine the concentration of a compound by examining the intensities of its correspond-

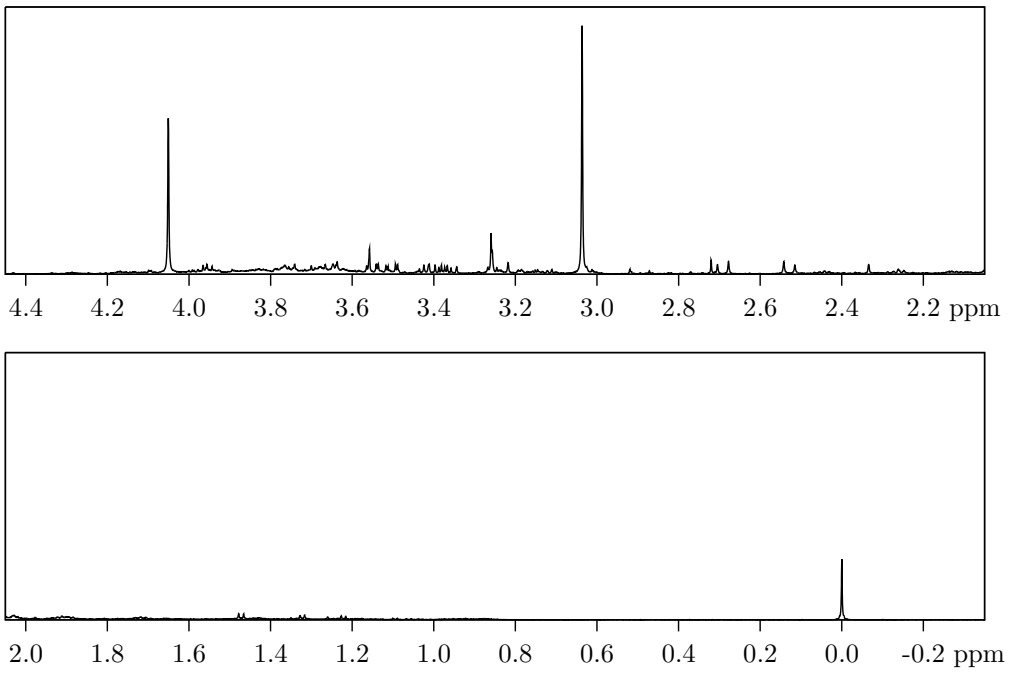

Figure 2.1: Example ^{1}H-NMR spectrum produced from a urine sample. Signal intensities are plotted based on frequency shifts with respect to the resonant frequency of the ^{1}H atom. The magnitude of these shifts is on the order of 10^{-6} times this frequency, thus they are given in parts per million (ppm).

ing peaks. This approach is called targeted profiling because it seeks to quantify specific metabolites (targets) in the spectrum.

An alternative approach to analyzing a spectrum that can produce meaningful measurements but avoids quantifying metabolite concentrations is called spectral binning. In this approach, the spectrum is divided into some number of frequency ranges called bins. Each bin becomes a variable with a value equal to the total signal intensity in the corresponding frequency range [32]. These intensities are directly related to the concentrations of compounds in the sample. This approach is simpler than targeted profiling because it avoids the intervention of the NMR expert to interpret the peaks. Because the measurements no longer correspond to metabolites, PIA cannot use them because it depends on metabolic pathway knowledge. Pathways describe the ways that specific metabolites interact with each other in various biological processes. It is not clear how to connect measurements from spectral binning with the metabolites on these pathways.

2.2 Metabolic Pathways

Metabolites interact with each other via chemical reactions that take place within an organism. These reactions convert one set of metabolites into another and they are typically catalyzed by a specific enzyme. An enzyme is a molecule that significantly increases the rate that a reaction takes place [37]. The rates of most reactions in the absence of an enzyme catalyst are too slow for those reactions to accomplish anything useful. Although the operation of an enzyme is complex and may perform several steps to complete a reaction, it is generally viewed as a single discrete step of converting one or more substrate molecules

into one or more products. Some reactions can occur in both directions, thus the role of substrates and products switch depending on the state of the organism.

The reactions involved in metabolism are organized into metabolic pathways. Pratt and Cornely discuss this organization along with several important metabolic pathways in detail [37]. A metabolic pathway is a set of reactions that are linked to each other through shared substrates and products. Within a pathway, these links indicate that a product of one reaction becomes a substrate in another. Pathways are defined based on the specific functions that they perform. There are two general categories of metabolic pathways: catabolic and anabolic. Catabolic pathways break large molecules down into smaller ones. These catabolic processes produce energy and the molecules needed to build other large molecules that an organism needs. The formation of large molecules from smaller ones is performed by anabolic pathways. Some pathways simultaneously perform functions that are both anabolic and catabolic.

One important pathway that is both anabolic and catabolic is the Citric Acid Cycle, also known as the Tricarboxylic Acid Cycle or the Krebs Cycle [37]. This pathway performs several important functions such as producing energy for the body. It also eliminates unneeded carbon atoms, in the form of CO_2, produced by catabolizing other molecules. The Citric Acid Cycle involves eight reaction equations. In these equations, a $\rightarrow$ indicates that a reaction is unidirectional. Here the set of substrates on the left of the arrow are always converted into the products on the right. Bidirectional reactions use $\leftrightarrow$ to indicate that the conversion of metabolites can occur in either direction. In order, the reactions of the Citric Acid Cycle pathway are:

1. $\underline{\text{Oxaloacetate}} + \text{Acetyl-CoA} + H_2O \rightarrow \underline{\text{Citrate}} + \text{CoASH}$
2. $\underline{\text{Citrate}} \leftrightarrow \underline{\text{Isocitrate}}$
3. $\underline{\text{Isocitrate}} + NAD^+ \rightarrow \underline{\alpha\text{-Ketoglutarate}} + NADH + CO_2$
4. $\underline{\alpha\text{-Ketoglutarate}} + \text{CoASH} + NAD^+ \rightarrow \underline{\text{Succinyl-CoA}} + CO_2 + NADH$
5. $\underline{\text{Succinyl-CoA}} + GDP + P_i \leftrightarrow \underline{\text{Succinate}} + GTP + \text{CoASH}$
6. $\underline{\text{Succinate}} + Q \leftrightarrow \underline{\text{Fumarate}} + QH_2$
7. $\underline{\text{Fumarate}} + H_2O \leftrightarrow \underline{\text{Malate}}$
8. $\underline{\text{Malate}} + NAD^+ \leftrightarrow \underline{\text{Oxaloacetate}} + NADH + H^+$

These reactions are linked together such that a product of one reaction becomes a substrate in the next. The links of the Citric Acid Cycle pathway are indicated by the underlined metabolites. For example, the first reaction produces citrate that is then consumed in the second reaction. As the name suggests, the Citric Acid Cycle also has the property that the pathway is cyclic. Thus, the oxaloacetate produced by the last reaction is reused as a substrate in the first reaction.

In addition to the connections within a pathway, there are also connections between pathways. The Citric Acid Cycle makes many connections to other pathways [37]. The Glycolysis pathway, for example, catabolizes glucose molecules to produce energy and pyruvate. These pyruvate molecules are converted to acetyl-CoA, a substrate in the first reaction of this pathway. Another example connection is via the metabolite α-ketoglutarate that is used to synthesize many amino acids. Catabolism of certain amino acids produces α-ketoglutarate as well. These kinds of connections imply that metabolism is a network of reactions and molecules instead of a collection of individual pathways. PIA derives a graph representation of the human metabolic network from a set of metabolic pathways from KEGG.

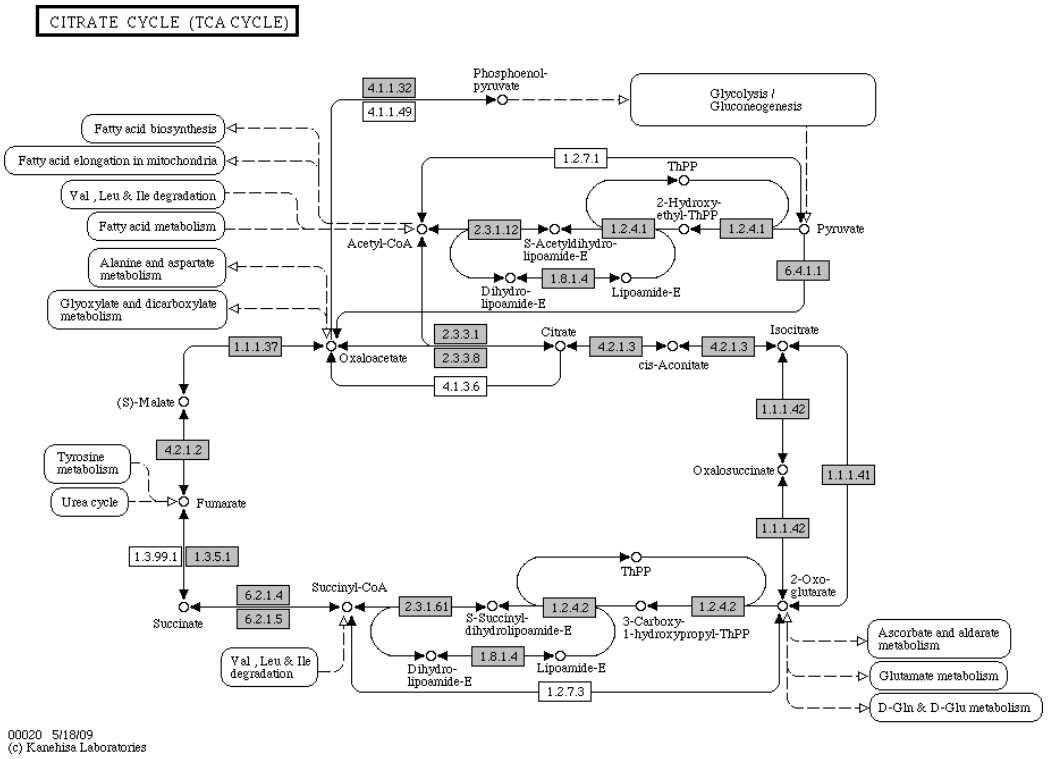

Figure 2.2: The KEGG representation of the human Citric Acid Cycle pathway [27]. Highlighted enzymes (rectangles) correspond to those that are encoded in the human genome.

2.3 Kyoto Encyclopedia of Genes and Genomes

KEGG is a collection of databases spanning a range of biological knowledge including metabolic compounds, enzymes, reactions and pathways [27]. The primary motivation for starting the project was to aid genome sequencing efforts. These efforts involve identifying a set of genes in an organism and then determining what function each of them performs. Many genes encode enzymes, thus it is desirable to know the identities of these enzymes to determine the reaction(s) that they catalyze. KEGG contains this type of functional information about genes in sequenced organisms. It also has information about the higher level organization of genes in the form of pathways that describe how genes interact with each other. One view of a metabolic pathway, for example, is as a sequence of genes that encode the enzymes catalyzing successive reaction steps. The idea is that this high level knowledge can help to assign functions to the genes of a new organism because different organisms have similar pathways. For example, it can help to verify functional assignments and identify missing assignments via incompletely assigned pathways [21]. PIA, however, is based on an alternative view of the structure and purpose of metabolic pathway knowledge.

In addition to the set of enzyme genes on each metabolic pathway, KEGG also contains the metabolites that participate as substrates and products in the reactions corresponding to those enzymes. PIA derives its metabolic pathway representation from these metabolite relations in contrast to a representation based on the interactions of enzyme-encoding genes. The developers of KEGG suggested a graph representation of metabolite relations where nodes represent metabolites and edges correspond to the reactions involving those metabolites [21]. They argued that this graph could be used to identify pathways present in an organism but did not consider a probabilistic interpretation of it in terms of conditional

independence statements.

The KEGG representation of the human Citric Acid Cycle pathway discussed in Section 2.2 is shown in Figure 2.2. From this pathway representation, it is clear that KEGG includes both enzymes and metabolites (as rectangles and circles respectively). Less clear is how to retrieve this information about a pathway. KEGG stores each metabolic pathway as a combination of pathways from the organisms for which it is known. After sequencing the genome of an organism and assigning functions to the enzyme genes, enzymes can be mapped onto these reference pathways to determine the organism-specific set of reactions for them [21]. The highlighted enzymes in Figure 2.2 indicate enzymes that are encoded in the human genome. As suggested by the enzymes not present in the human genome, other organisms accomplish the function of this pathway with a different set of enzymes. Fortunately, the human genome is already known, thus the information is available to help determine what parts of the KEGG pathways to use in analyzing human metabolic profile data.

There are several reasons that we selected KEGG for the development of PIA. First, the database originated in 1995 [26]. Since PIA depends on metabolic pathways, we are particularly interested in this section of KEGG. Metabolic pathways were an early focus of attention in the development of KEGG [27]. According to the developers, these pathways were the best organized pathways in KEGG early in its development [21]. Another major reason is the ease with which KEGG data can be incorporated into PIA. Information describing the needed components of the KEGG graphical pathway diagrams is available in an easily parsed format. KEGG provides all the necessary data files instead of a limited set of software programs or a narrow interface to access its information. This makes it easy to take advantage of KEGG for tasks that were not envisioned by its creators. As noted above, PIA represents a novel application metabolic pathways to develop a classifier that diagnoses a disease from a metabolic profile.

2.4 Machine Learning

Classification is a fundamental problem addressed by the field of machine learning [22]. We are interested in this area because the disease diagnosis task is an instance of a classification problem. A classification problem involves objects that are described by a set of features and that belong to one of several categories or classes. The goal in machine learning is to produce a program, called a classifier, that is able to determine the correct category for an object based only on its feature values. In the context of our disease diagnosis problem, the objects are patients that are described by their metabolite concentration values. Thus, there is one feature corresponding to each of the metabolites measured in a metabolic profile. A diagnosis tool based on a machine learning classifier predicts whether or not an undiagnosed patient has a disease given their metabolic profile. The metabolite concentrations are the input to this program and the output is a diagnosis. Naturally, the more often these diagnoses are correct, the better the underlying classifier is.

Classifiers are based on models that are derived from a set of example instances, each correctly labeled with its corresponding class. The set of labeled example instances is called a training set and the models are derived by a process called learning. This process is called supervised learning because it depends on knowledge of the correct classes. As noted, we want a classifier that accurately predicts the correct classes of future instances. Because future data is not available by definition, we can estimate the performance of a learned classifier by dividing the available data into two data sets called the training set and the test set. The training set is used to learn a classifier that is then evaluated on the test set where the correct classes of these instances are hidden. The predicted classes are compared with the correct classes to measure accuracy as a percentage of correctly classified instances. The learning and prediction processes vary for different machine learning algorithms. Several

commonly used machine learning algorithms that we consider for our disease diagnosis task are described below.

2.4.1 Algorithms

This section discusses the machine learning algorithms that we use to evaluate the ideas presented in this thesis. Since our diagnosis problem is a binary classification task (a task with two classes), we describe the algorithms in this context. These algorithms can be generalized to handle problems with more than two classes but this typically introduces additional complexity. Focusing on the simpler case simplifies the discussion and makes it more relevant to our specific problem. The following standard machine learning algorithms are used in this thesis.

- **Support Vector Machines (SVM)** are linear classifiers that seek a linear function of the attributes $f(x_r) = w_0 + w_1 x_{r1} + w_2 x_{r2} + \ldots + w_p x_{rp}$ that can separate the two classes from each other. In other words, points corresponding to metabolic profiles of positive patients should occur on one side of the hyperplane defined by this function and negative patients on the other side. If such a hyperplane exists then the data points are linearly separable. In this case, there are potentially many hyperplanes able to separate the classes. For any given hyperplane, one or more points from each class are closer to it than the others in the same class. These points are called support vectors and their distance from the hyperplane is called the margin. An SVM chooses the hyperplane that maximizes the size of the margin based on the idea that this choice yields the best generalization (see Section 2.4.4) to future points.

  In general, a hyperplane that perfectly separates the training points may not exist. One approach to this problem is to map the data into a higher dimensional feature space using kernel functions where it might become linearly separable. An example of a commonly used kernel function is the radial basis kernel defined as $K(x_q, x_r) = \exp(-\gamma \|x_q - x_r\|^2)$ for $\gamma > 0$. The function $K(x_q, x_r)$ is applied to all pairs of training data points x_q and x_r (not necessarily distinct) prior to learning. Since misclassifications may occur regardless of the feature space, SVMs are generalized to allow points to occur on the wrong side of the hyperplane. The generalized approach imposes a penalty for each misclassified point that increases with its distance from the hyperplane. These penalties are incorporated into a constrained optimization problem that simultaneously maximizes the margin and minimizes the misclassification penalties. This problem is a convex quadratic program, thus there is an efficient solution for the optimal coefficients $w = (w_0, w_1, \ldots, w_p)^T$. After training, the class of an undiagnosed patient q is assigned as positive if $f(x_q) \geq 0$ and as negative otherwise. Many authors describe SVMs in greater detail including Shawe-Taylor and Cristianini [15], Bishop [7] and Hastie et al. [22].

- **Decision Trees (DT)** apply a sequence of tests to the attribute values of an instance to determine a classification. The simplest decision tree that always predicts a single class is called a leaf. More general trees start at a root specifying a test to apply and having a DT associated with each possible outcome. Tests for continuous attribute values, such as metabolite concentrations, consist of an attribute and a threshold. Given an instance, the root test of a DT is applied by comparing an attribute value x_{ri} to a threshold t. The comparison $x_{ri} \geq t$ is either true or false, thus there are two possible outcomes that each branch to an associated DT. The appropriate DT is selected for the instance and further tests are applied in this manner until a leaf predicting a class is reached. DT models are advantageous because they are more easily interpreted than other models, potentially revealing insights into the structure of a classification problem.

The specific algorithm for learning DT models that we consider here is C4.5 [38]. Given a training set, the algorithm considers all possible ways to test each attribute by considering each attribute value as a threshold. Because a specific test implicitly divides the training data into two subsets, the algorithm computes a measure of the class purity of the subsets to determine the best possible test. This greedy test selection mechanism is recursively applied to learn the outcome-specific DTs from the corresponding training subsets. Learning eventually terminates with a leaf when these subsets contain instances from only a single class. A pruning step is then applied to simplify the structure of the learned DT and to allow it to better generalize (see Section 2.4.4) to future instances. Quinlan discusses C4.5 in greater detail as well as general issues related to DT learning [38].

- **Graphical models** compactly represent multivariate probability distributions by explicitly incorporating independence statements about them [30]. This thesis focuses on undirected graphical models only. In these models, independence statements are represented by an undirected graph structure where nodes correspond to variables and edges represent direct dependencies. The absence of an edge between two nodes implies a set of conditional independence statements relating the corresponding variables that can be determined by examining the graph. The compactness of the representation is derived from the sparsity of the graph structure. Fewer parameters are required to represent a distribution over a graph with a small number of edges than one with many edges. Given a parameterized distribution, we can create Bayesian classifiers as described in Section 1.4 by representing the likelihood distribution $P(x_r \mid c_r)$.

  Several standard machine learning models fit naturally into the framework of graphical models. The Naïve Bayes (NB) model assumes that the variables in a distribution are independent of each other given only the class c_r [30]. Tree-augmented Naïve Bayes (TAN) and Multi-TAN (MTAN) allow a limited set of conditional independence statements that are captured by tree-structured graphs [20]. We also consider a full dependence (Full) model that assumes every variable is dependent on every other variable in a distribution. Our PIA system is based on an interpretation of metabolic pathways as probabilistic independence statements about metabolite concentrations. This interpretation allows us to treat the pathways as the underlying graph structure for distributions over metabolic profiles. Chapter 4 shows how to implement these models using a specific type of undirected graphical model called a GMRF.

These algorithms share the common assumption that the training and testing sets contain no missing measurements. When working with metabolic profile data, this is problematic for two reasons. Occasionally it is not possible to determine the concentration of a metabolite from an NMR spectrum. Also, metabolic profiles only measure a subset of the metabolites in an organism, thus some potentially important metabolites are never measured. These unmeasured metabolites correspond to latent variables in the models that we consider. As we will see in Chapter 5, models that account for these latent variables can be much simpler than those that do not. We employ a general approach commonly used in machine learning called Expectation Maximization (EM) to address the problem of missing values.

2.4.2 Missing Values

EM is a general technique for handling missing values that works with models that represent probability distributions over a set of variables that are not always measured [16]. The graphical models described above are an example of this type of model. Distributions are described by a set of parameters that are learned from a training data set. The procedure starts with some guess of the values for the parameters (chosen randomly for example) and maintains a set of current parameters throughout. EM is a two step procedure.

The expectation step uses the current parameters to compute the expected value for every missing value in the training data given the observed values from the corresponding instances. These computed values are treated as if they are the actual values for the missing measurements. In a sense, this step fills in the missing values in the data to produce a data set with no missing values. The maximization step takes the full training data and learns a new set of parameters according to some learning algorithm. These new parameters are then treated as the current parameters and the EM procedure is repeated until the current parameters converge to a set of fixed values. The final parameters are not necessarily the optimal values but they typically perform well in practice.

2.4.3 Cross Validation

As noted above, the performance of a classifier is determined by its ability to accurately classify future instances. We simulate these instances using data we already have with a technique called k-fold cross validation (CV) [22]. The first step of the procedure is to partition the data set into k subsets, called folds, that are approximately equal in the number of instances that they contain. One of the folds is held aside and the remaining $k-1$ folds are used as a training set for the machine learning algorithm. The held out fold is then used as the test set. The classes of these instances are kept hidden and the learned classifier is evaluated on them, thus they effectively represent future instances.

This process is repeated for each of the other folds, yielding an accuracy rate on each of the k test sets. Averaging these values gives an estimate of the performance of the learned classifier applied to every instance in the data set. These fold-specific accuracy values can be used to compare the performance of two learning algorithms as well. Assuming that the folds are the same for both algorithms and treating fold accuracy as a random variable, we can use a paired t-test to determine if the mean fold accuracy values are significantly different for the two algorithms. Another advantage of cross validation is that it makes efficient use of the data. Every instance gets used as a test instance once and every training set uses most of the available instances, approximately $(k-1)/k$ of them [7]. Leave one out CV (LOOCV) is a special case of CV that sets $k = n$, the number of instances in the data set. This procedure makes the most efficient use of the available instances by maximizing the size of each training set but it requires more computational effort than smaller values of k would.

2.4.4 Overfitting

Overfitting is a phenomenon that affects the classification performance of machine learning algorithms. A good model is one that performs well on future instances or, more concretely, accurately classifies instances in the test set [22]. Machine learning algorithms are applied to instances in the training set to infer something about a mechanism of interest. In addition to characterizing this mechanism, the training instances are assumed to contain some noise as well. Overfitting occurs when the model is more complicated than the underlying mechanism, allowing the learning algorithm to model too much of the noise. An overly-complicated model is able to more accurately classify training set instances at the expense of performance on the test set [22]. Thus, a simple measure of the degree of overfitting exhibited by a model is its accuracy at classifying instances in the training set.

In our case, we assume that there is some mechanism governing normal metabolism that determines urinary metabolite concentrations in patients that do not have cachexia and a different mechanism for those that do. Random noise is introduced in a number of ways. Metabolism varies according to the diet, age and sex of a patient for example. Quantification of metabolite concentrations by NMR spectroscopy is imperfect, thus it is a source of noise as well. One way to overcome the effect of noise is to increase the number of instances in the

training set. Since we cannot simply increase the size of our data set, we consider varying model complexity instead.

Unfortunately, the appropriate level of complexity is unknown. Classifier performance generally improves with increasing model complexity up to a certain point where classification accuracy on future instances decreases as overfitting increases [22]. We consider models over a range of complexity levels from those with few parameters to models with as many parameters as possible under the assumption that the class conditional distributions follow multivariate normal distributions. PIA is an attempt to find the optimal complexity level that balances the complexity of the disease mechanism with minimal overfitting. It accounts for mechanisms determined by known metabolic pathways but no others. The following chapters describe how PIA is implemented and discuss its performance relative to other models.

Chapter 3

Related Work

The development of PIA combines several threads of previous research. First, machine learning techniques have already been applied to metabolic profile data with the specific goal of developing disease diagnosis tools. Only a limited amount of biological knowledge was used in these previous applications however. To our knowledge, metabolic pathways were not incorporated in any of them. In addition to its learning component, PIA interprets metabolic pathways in graph theoretic terms. Previous work has examined the properties of graph representations of metabolic networks for various organisms. From this work, we derive our technique for converting a set of pathways into a graph structure.

This chapter summarizes this relevant work as well as a few other existing ideas important to the development of PIA. These include the idea of currency metabolites that we use to simplify the models that PIA employs. Currency metabolites are assumed to be available in sufficiently high quantity in the body that they do not influence the concentrations of the other compounds that react with them. Understanding this related work is necessary to understand the development of the PIA system.

3.1 Metabolic Profiles and Machine Learning

Baumgartner et al. applied machine learning algorithms to diagnose two metabolic diseases from metabolic profiles [6]. The first disease, called phenylketonuria (PKU), involves a deficiency of the enzyme that catalyzes a reaction producing tyrosine from phenylalanine. This defect causes concentrations of phenylalanine to increase and tyrosine to decrease in affected patients. The disease mechanism leads to simple diagnostic rules based on these concentration values [9, 10]. The other disease Baumgartner et al. considered was medium-chain acyl-CoA dehydrogenase deficiency (MCADD). Fatty acids are not metabolized correctly in patients with MCADD resulting in higher concentrations of these metabolites. For example, the concentration of C8-carnitine is particularly increased in MCADD patients and its value is the basis of similar diagnostic rules for this disease [44]. Intuitively, both diseases provide some diagnostic information in the metabolite concentrations of affected patients. This information should allow machine learning algorithms to accurately classify healthy and diseased patients.

To evaluate this hypothesis, Baumgartner et al. used metabolic profiles derived from blood samples taken from newborns [6]. Their data set contains 43 PKU patients, 63 MCADD patients and 1,241 patients with no known metabolic diseases. The newborn metabolic profiles quantify the concentrations of 14 amino acid and 29 fatty acid metabolites. Machine learning algorithms were applied to the amino acids for PKU and the fatty acids for MCADD because of the nature of these diseases. Thus, some expert knowledge was used to do feature selection initially. Two classification tasks were considered: PKU

vs. healthy and MCADD vs. healthy. Baumgartner et al. employed SVMs and DTs as well as the machine learning algorithms Partial Least Squares Discriminant Analysis (PLS-DA), Logistic Regression (LR), k Nearest Neighbors (k-NN), Artificial Neural Networks (ANN). In all but two cases, the accuracy of the machine learning algorithms in 10 fold CV exceeded 99%. When LR and k-NN are applied in the MCADD task, their accuracies each exceeded 98%. The authors do not state whether any of the differences are statistically significant.

Baumgartner et al. expanded on their newborn disease diagnosis work by considering additional features in their machine learning diagnosis tools [5]. In addition to individual metabolite concentration values, diagnosis rules have been developed based on ratios of metabolite concentration values. The authors considered these ratios as a form of expert knowledge and treated their values as distinct variables that are used as inputs to the machine learning algorithms. To classify MCADD patients, for example, the value of the ratio between the concentrations of C8 carnitine and C10:1 carnitine was an input variable in addition to the individual metabolite concentrations.

They tested this idea using a similar data set to the one used above. Instead of PKU, they considered a more general disease called phenylalanine hydroxylase deficiency (PAHD) that includes PKU as a special case. Thus, the data set contains the 43 PKU patients plus 51 non-PKU patients for a total of 94 patients in the PAHD class. The 63 MCADD patients are also in this data set. Finally, they added 22 patients with a third disease called 3-methylcrotonyl CoA carboxylase deficiency (3-MCCD). The same number of normal patients were used again and a similar division into three disease classification tasks was used.

Two machine learning algorithms were considered: LR and DT (C4.5 algorithm). LR was given the additional expert features but DT was not. Results from 10 fold CV indicate that the accuracy exceeded 99% in all cases as before. It appears that, with the expert knowledge, LR classifiers performed significantly better than the DT classifier on the positive patients. Again, the authors do not address the statistical significance of any of their results.

One major problem with the work of Baumgartner et al. [6, 5] is that they evaluated the machine learning classifiers in isolation. The authors acknowledge that, for the diseases they considered, current diagnosis methods are based on rules applied to the concentrations of specific diagnostic metabolites. They did not compare their machine learning results with these methods nor did they describe them. MCADD, for example, can be diagnosed from the concentrations of marker metabolites or through genetic analysis to identify the mutation causing the enzyme defect [10]. In order to justify the use of machine learning algorithms to diagnose a disease, something must be gained. The gain does not necessarily have to come from improved classification accuracy. Classifiers produced by machine learning techniques may be faster, require less expert intervention, cost less or use biological samples that are less invasive to acquire. Baumgartner et al. motivate their work based on the large number of samples to be analyzed [6] but this only suggests computational analysis, not necessarily a machine learning approach. They should address the question of why it is infeasible to implement the established diagnostic rules directly in a computer program.

Mahadevan et al. also investigated the efficacy of machine learning algorithms applied to metabolic profile data [32]. They considered two machine learning tasks, one of them diagnostic in nature. The tasks were identifying pneumonia patients and distinguishing male and female from urinary metabolic profile data. Mahadevan et al. use a data set containing 59 patients with pneumonia and a total of 352 normal patients. Of these healthy patients, 194 of them are female and 158 are male. They evaluated two machine learning algorithms on the two tasks: SVMs and PLS-DA.

The authors considered two views of the NMR spectra in their data set. One was based on a spectral binning approach that yields a set of 360 bins (features) per patient. The other was a targeted profiling approach that produces a metabolic profile quantifying 82 metabolite concentrations for each patient. The latter approach has the advantage of allowing for drug metabolites to be removed before analysis. This is necessary for patients

in the data set diagnosed through other means that are already receiving drugs to treat their pneumonia. Naturally, patients without pneumonia are not likely to take these drugs. Knowing the concentrations of drug metabolites is highly indicative of the disease state of a patient but a patient cannot begin treatment until a diagnosis is made. Therefore, the input to a disease classifier will not have this additional information from drug metabolite concentrations if it is used as a diagnosis tool in a clinical setting. The authors state that their work is intended to evaluate learning algorithms on metabolic profile data and it is not presented as a new method of diagnosis.

The authors performed 4 fold CV and LOOCV to evaluate machine learning algorithms on their data set. In both cases, results based on spectral binning were slightly better than those based on targeted profiling. SVMs slightly outperformed PLS-DA for this task in all cases as well. The authors do not discuss the statistical significance of their results. Their work also lacks a discussion of existing diagnosis methods for pneumonia.

Our work on PIA is an attempt to develop classification algorithms for metabolic profiles that outperform existing machine learning methods and that are faster and less invasive than existing approaches for making diagnoses. To do this, we attempt to take advantage of metabolic pathway knowledge that describes relationships among the metabolites in a metabolic profile. Incorporating pathways into computational analysis is a challenge in itself however. Addressing this challenge requires an appropriate representation for metabolic pathways.

3.2 Graph Representations of Metabolic Pathways

Developing a classifier using metabolic pathways requires a representation suitable for computational analysis. As described above, metabolic pathways are composed of chemical reaction equations linked together via product metabolites that become substrates in subsequent reactions. For several reasons, the most natural approach is to represent pathways using graph structures. Graphs are well understood theoretically, easy to incorporate into computer programs and fit into the framework of probabilistic graphical models. In this framework, the nodes of a graph correspond to variables and an edge between a pair of nodes indicates a direct dependence between the corresponding variables [30]. A significant amount of work exists that has sought to represent metabolic pathways using graphs in various ways and to analyze the structural properties of these graphs. This section reviews several of these representations. We focus on representations that explicitly include metabolites as nodes because the variables in a metabolic profile data set correspond to metabolite concentrations.

To illustrate the various graph representations and related ideas we use an example metabolic reaction taken from the Citric Acid Cycle pathway as described in Section 2.2. The reaction is the irreversible third step of this pathway that involves five metabolites (2 substrates and 3 products). The reaction equation is

$$\text{Isocitrate} + \text{NAD}^+ \rightarrow \alpha\text{-Ketoglutarate} + \text{CO}_2 + \text{NADH}$$

and it is catalyzed by an enzyme called isocitrate dehydrogenase with Enzyme Commission number 1.1.1.41 [37]. We show how to covert this reaction into a graph for each of the graph representations considered in this section. Given the entire set of metabolic reactions from a pathway or multiple pathways, a combined graph representing them is constructed in a natural way. The set of nodes in the combined graph is the set of unique nodes in all reaction graphs. The set of edges is the union over all edges. Two nodes are unique if and only if they correspond to different metabolites. This section analyzes the advantages and disadvantages of various graph representations of metabolic pathways from the perspective of PIA. We argue for using a particularly simple undirected graph representation that captures

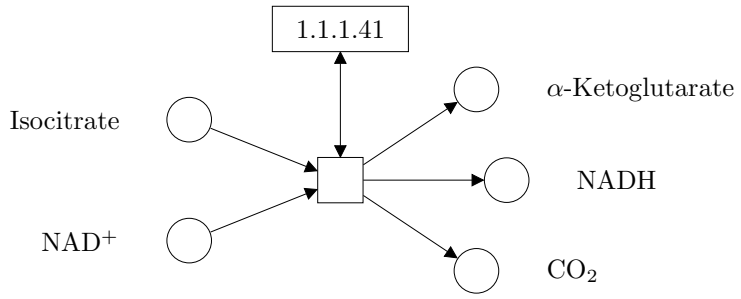

Figure 3.1: Graph representation of the third reaction of the Citric Acid Cycle pathway proposed by Jeong et al. [25].

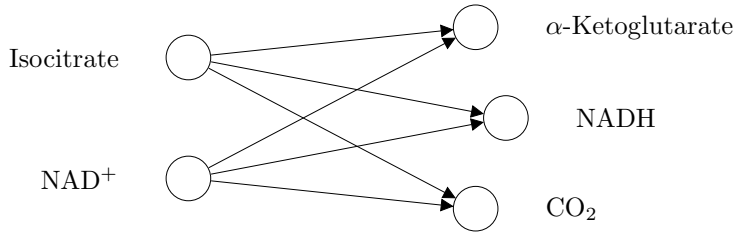

Figure 3.2: Graph representation of the third reaction of the Citric Acid Cycle pathway proposed by Ma and Zeng [31].

an intuitively meaningful set of conditional independence statements about distributions of metabolite concentrations.

An early study of the statistical properties of metabolic networks was conducted by Jeong et al. [25]. They analyzed the graph structures of 43 organisms and found that they exhibit similar degree distribution and average path length. To facilitate this analysis, Jeong et al. construct a directed graph representation of the metabolism of each organism where metabolism is composed of the set of metabolic reactions that an organism can perform. There are three types of nodes in their representation: metabolite, enzyme and reaction. In this representation, our example reaction would correspond to the graph shown in Figure 3.1. If the reaction were reversible, then the edges adjacent to the metabolites would become bidirected. Directed edges make sense for Jeong et al. because they are interested in path lengths between pairs of metabolites or, equivalently, the number of steps needed to covert one metabolite into another. This conversion cannot happen by going against the direction of a reaction. The representation of Jeong et al. includes metabolites that are quantified by metabolic profiles but their representation also contains nodes that do not correspond to metabolite concentrations.

Ma and Zeng proposed a similar graph representation that includes metabolite nodes only [31]. This representation places edges between every pair of nodes where one is a substrate and the other is a product in the same metabolic reaction. For irreversible reactions, these edges are directed from substrate node to product node. If a reaction can operate in both directions then the corresponding edges are undirected. The graph of the example Citric Acid Cycle reaction is shown in Figure 3.2. Note that this representation does not connect two substrates (or two products) that occur in the same reaction together unless one is a substrate and the other is a product in some other reaction. Huss and Holme use this method

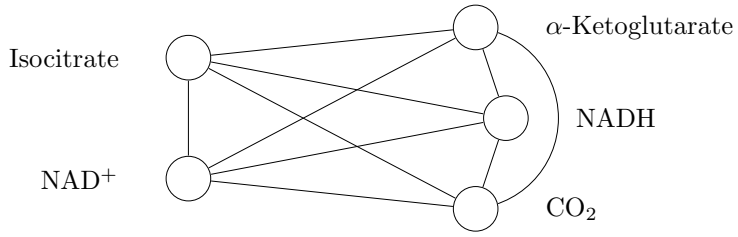

Figure 3.3: Graph representation of the third reaction of the Citric Acid Cycle pathway proposed by Wagner and Fell [45].

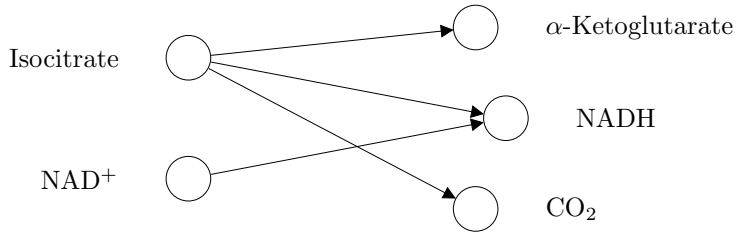

Figure 3.4: Graph representation of the third reaction of the Citric Acid Cycle pathway proposed by Arita [1, 2].

for constructing graphs from metabolic networks except that all edges are undirected in their representation [23]. In contrast to the mixed directionality graph of Ma and Zeng, this undirected version easily fits into the framework of probabilistic graphical models. A natural probabilistic interpretation applies to this particular graph structure: the concentration of a metabolite is independent of all others given the concentrations of all metabolites that are used to produce or consume it in a reaction. This interpretation has a major flaw however because the metabolites that a substrate reacts with also influence its concentration. If these metabolites were removed then the concentration of that substrate is directly affected because the reaction could no longer occur.

A slight variation on the graph representation of Huss and Holme that yields a more sensible interpretation was proposed by Wagner and Fell [45]. Their graph construction also produces an undirected graph. The resulting graph differs from that of Huss and Holme because it does not distinguish substrates and products of a chemical reaction. Instead, any two compounds that participate in the same reaction are connected via an edge. The graph constructed from the example reaction following this method is shown in Figure 3.3. This metabolic graph representation gives the desired probabilistic interpretation. In this case, the concentration of a metabolite is independent of all others given the concentrations of the metabolites participating in a common reaction. No other metabolites could influence the production of consumption of the metabolite in question. Influences on this node are effectively blocked by its neighbors in the graph. We select the Wagner and Fell (WF) representation as the basis for the PIA system.

Although the selected structure yields a convenient interpretation from a machine learning perspective, it loses much information about the pathways themselves. Arita developed a more biologically realistic graph representation that attempts to more accurately represent the operation of metabolic pathways [1, 2]. The graph construction is based on a specific

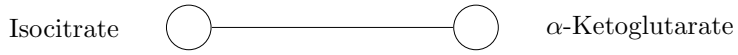

Figure 3.5: Graph representation of the third reaction of the Citric Acid Cycle pathway after removing the currency metabolites NADH, NAD^+ and CO_2.

definition of a pathway. According to this definition, a pathway exists between two metabolites if and only if an atom from one can reach the other through a sequence of reactions. To identify pathways, it is necessary to know how metabolites are changed by metabolic reactions. This leads to atomic mappings that describe how each atom in a substrate appears in a product of a reaction. Directed edges are derived from these mappings that connect a substrate and product if and only if they share an atom. This produces a more sparse graph structure with fewer edges as shown for the example reaction in Figure 3.4. The representation does not capture a meaningful set of independence statements however. To increase the sparsity of the WF graph representation, PIA takes advantage of the idea of currency metabolites discussed in the next section.

3.3 Currency Metabolites

Currency metabolites are a class of metabolites defined by their prevalence in an organism. An exact definition of what metabolites are in this class does not exist, although many authors have discussed the idea of currency metabolites. These metabolites are assumed to be available in sufficient quantities in the body that they do not influence the occurrence of any reactions. Also, they tend to occur in many more reactions than the average metabolite [23]. Thus, they will result in nodes of high degree in any of the graph representations considered above. Structural analyzes of metabolic graphs tend to underestimate properties such as average path length when currency metabolites are included because they are connected to so many other metabolites. Ma and Zeng argue that these currency metabolites should be removed from metabolic graphs because paths that pass through them do not correspond to valid metabolic pathways [31]. There is no agreement on the metabolites to remove, although the currency metabolite lists that appear in the literature are similar.

Several authors simply state a set of metabolites that they consider to fit the definition of currency metabolite. Ma and Zeng consider defining currency metabolites on a per-reaction basis [31]. Huss and Holme implement a graph clustering approach to identify currency metabolites computationally [23]. The idea here is that the metabolic graph can be grouped into clusters of nodes. Within a cluster, the nodes are densely connected to each other with many edges but between clusters there are much fewer edges. The presence of currency metabolites distorts the clustering however. Their approach iteratively removes metabolites attempting to increase a score function based on how well the graph can be clustered. The result is a list of metabolites that they define as currency metabolites. Table 3.1 shows this currency metabolite list as well as the lists of several other authors[1]. Because PIA uses the WF graph representation, we will use the corresponding currency metabolite list as well.

We believe that PIA is justified in ignoring these currency metabolites. If a metabolite is always available in sufficient quantity that it does not affect the occurrence of any reactions, then the concentrations of other metabolites do not depend on it. This is because knowing the concentration of this metabolite reveals nothing about the others. We have already stipulated that its concentration is high regardless of the disease state of an organism. Ignoring these currency metabolites simplifies the models that PIA employs. Among the

[1]Note that KEGG does not differentiate between NAD^+ and NAD and between $NADP^+$ and NADP.

Wagner and Fell	Ma and Zeng	Huss and Holme
ATP	ATP	ATP
ADP	ADP	ADP
NAD	NAD^+	NAD^+
NADH	NADH	NADH
NADP		$NADP^+$
NADPH		NADPH
NH_3	NH_3	
CO_2	CO_2	
	O_2	O_2
	H_2O	H_2O
PP_i		
Thioredoxin		

Table 3.1: Currency metabolites proposed by Wagner and Fell [45], Ma and Zeng [31] and Huss and Holme [23] that are also in KEGG.

common currency metabolites that appear in the literature are NADH, NAD^+ and CO_2. These three metabolites are all involved in the example Citric Acid Cycle reaction used in Section 3.2. If we consider ignoring these in the graph construction procedure, the result is the much simpler graph shown in Figure 3.5. In this case, following the WF graph, the graph is reduced in size by three nodes and nine edges. The result is a smaller number of parameters that PIA must learn. Although this example is an extreme case, incorporating the additional biological knowledge of currency metabolites results in a reduction in the size of the graphs used by PIA.

Chapter 4

Markov Random Fields

This chapter discusses the theory of GMRF models, the type of undirected graphical model that PIA uses to incorporate metabolic pathway knowledge into a disease classifier. The standard machine learning models NB, TAN, MTAN and Full have natural interpretations as GMRF models that we describe here as well. There are two basic GMRF learning problems that this chapter considers: learning the parameters of a model given a graph structure and learning the underlying structure itself. We present algorithms that solve both of these problems. Finally, this chapter shows how to use GMRF models to represent class-conditional distributions of the form $P(x_r \mid c_r)$ and how to use them to produce a classifier.

Most of the material in this chapter summarizes existing work in machine learning. In particular, the first two sections of this chapter summarize material from Koller and Friedman [30]. Later sections are derived from a variety of sources that are cited where appropriate.

4.1 Basic Graph Theory

A graph structure $G = (V, E)$ consists of a set of nodes $V = \{1, 2, \ldots, p\}$ and a set of edges $E \subseteq V \times V$ that connect pairs of nodes. An example of a graph with 7 nodes and 10 edges is shown in Figure 4.1. In this thesis, we consider only undirected graphs where the order of nodes in an edge is irrelevant. Specifically, if nodes u and v are connected, then there is an edge $e = \{u, v\}$ and it is equivalent to the edge $\{v, u\}$. If there is an edge $e = \{u, v\}$ then u and v are adjacent to each other and the edge e is said to be incident to both u and v. We occasionally assign a weight value to edges in a graph via a function $w : E \mapsto \mathbb{R}$.

A sequence of nodes $v_1, v_2, \ldots, v_k$ is called a path if and only if successive nodes in the sequence are joined via edges. Thus, $\{v_1, v_2\} \in E, \{v_2, v_3\} \in E, \ldots, \{v_{k-1}, v_k\} \in E$. The length of a path is the number of edges it contains. The sequence of nodes 1, 2, 5, 6 and 4 in the graph in Figure 4.1 is a path of length 4. If $v_1 = v_k$, then the path is called a cycle. The graph in Figure 4.1 contains the cycle 2, 3, 6 and 5 among others. A graph with no cycles is called a tree. An important concept related to paths is called separation. A path from node a to b is separated by a node c if and only if c occurs on the path. More generally, two sets of nodes A and B are separated by a third set C if and only if every path from a node $a \in A$ to a node $b \in B$ contains a node $c \in C$.

The set of nodes denoted by $N(v)$ is called the neighborhood of the node v. It contains all nodes u such that u and v are adjacent and no others. In the graph in Figure 4.1, the neighborhood of node 3, $N(3)$, consists of nodes 2, 4, 5 and 6 for example. The degree of a node v, denoted $\deg(v)$, is the number of nodes that it is adjacent to or, equivalently, the size of its neighborhood $|N(v)|$. In Figure 4.1, $\deg(3) = 4$ for example.

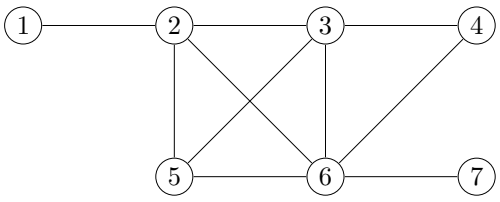

Figure 4.1: Example graph structure consisting of the nodes $V = \{1, 2, 3, 4, 5, 6, 7\}$ and the edges $E = \{\{1, 2\}, \{2, 3\}, \{2, 5\}, \{2, 6\}, \{3, 4\}, \{3, 5\}, \{3, 6\}, \{4, 6\}, \{5, 6\}, \{6, 7\}\}$.

A clique is a set of nodes $\{v_1, v_2, \ldots, v_k\}$ with the property that every pair of nodes in the set is adjacent. A maximal clique is one to which no additional nodes can be added without losing this property. The graph in Figure 4.1 contains several cliques including $\{2, 3, 5\}$ and $\{2, 3, 5, 6\}$ for example. The latter set of nodes is a maximal clique but the former is not.

One important use of graph structures is to model properties of joint probability distributions over multiple variables. As suggested in Section 1.3, this thesis is concerned with models based on undirected graph structures. These structures are the basis of a general class of graphical model called Markov Random Fields. In the following sections we describe how these models represent probability distributions and how to develop classifiers based on them.

4.2 Markov Random Fields

A Markov Random Field (MRF) represents a joint probability distribution over a set of random variables $\{x_1, x_2, \ldots, x_p\}$. This representation explicitly incorporates conditional independence statements about the distribution in the form of an undirected graph. In this graph $G = (V, E)$, each node $v \in V$ corresponds to one variable x_v in the distribution and every variable has a corresponding node. An edge between two nodes in the graph indicates a direct dependence between the corresponding variables. Indirect dependencies are captured via paths over multiple edges in the graph. Intuitively, observing the state of a variable will reveal information about the states of variables that depend on it. In a sense, information about other variables flows through paths in the graph structure from this variable.

This information flow is driven by observed variables but it is limited by them as well. The mechanism is formalized based on the notion of conditional independence of variables. Here, two variables x_i and x_j are independent of each other given the value of some other variable x_k if and only if $P(x_i, x_j \mid x_k) = P(x_i \mid x_k)P(x_j \mid x_k)$. This conditional independence is denoted as $(x_i \perp x_j \mid x_k)$. Observing the value of x_k blocks the flow of information, if any, between x_i and x_j. The idea of conditional independence can be generalized to sets of variables x_I, x_J and x_K where the subscripts I, J and K denote disjoint subsets of V. Conditional independence statements about the variables are explicitly encoded in the graph structure of an MRF.

There are several ways to characterize the independence statements made by an MRF graph structure. To do this, we use the notation $x_{\overline{I}}$ as a shorthand for $x_{V-I} = \{x_i \mid i \notin I\}$, the set of variables with indices not in the set I. The characterizations are as follows.

- **Pairwise independence** indicates that a pair of variables x_u and x_v are conditionally independent given all others if and only if the corresponding nodes u and v are not adjacent. The set of independencies in this case is $(x_u \perp x_v \mid x_{\overline{\{u,v\}}})$ for all $\{u, v\} \notin E$.

For example, according to the graph in Figure 4.1, the variables x_2 and x_4 are pairwise independent because nodes 2 and 4 are not adjacent.

- **Local independence** means that a variable x_v is independent of all others given those variables corresponding to nodes in its neighborhood $N(v)$. The set of independence statements is $(x_v \perp x_{\overline{N(v)\cup\{v\}}} \mid x_{N(v)})$ for all $v \in V$. For example, according to the graph in Figure 4.1, the variable x_3 is independent of x_1 and x_7 given the variables neighboring x_3: x_2, x_4, x_5 and x_6.

- **Global independence** states that two sets of variables x_I and x_J are independent of each other given the set of variables x_K if and only if every path between a node in I and a node in J in G passes through a node in the set K. Thus $(x_I \perp x_J \mid x_K)$ for all sets of nodes I, J and K such that the nodes in K separate those in I and J. For example, variables $x_{\{1,2,5\}}$ are independent of $x_{\{4,7\}}$ given $x_{\{3,6\}}$ according to the graph in Figure 4.1.

Each of the three types of independence statements implies the other two for a graph structure if the joint distribution it represents is positive. This means that $P(x_1, x_2, \ldots, x_p) > 0$ for all values of the variables $x_1, x_2, \ldots, x_p$. We employ a specific parameterization of MRF models, described below in Section 4.3, that ensures that this requirement is met.

Up to this point, we have focused on the qualitative properties of probability distributions. MRF models and other graphical models are useful because they separate these properties from the quantitative properties represented by a specific parameterization. This allows us to use expert domain knowledge to determine the structure of a distribution and then use machine learning algorithms to determine the parameters that define it. This idea is the basis of the PIA system. Because PIA must handle continuous variables from metabolic profile measurements, we use a specific kind of MRF model that represents a multivariate normal (or Gaussian) probability distribution.

4.3 Gaussian Markov Random Fields

Suppose that the variables $x_1, x_2, \ldots, x_p$ follow a joint normal probability distribution. Thus, for the vector $x = (x_1, x_2, \ldots, x_p)^T$, we have the probability density function

$$P(x \mid \mu, \Sigma) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right] \tag{4.1}$$

with mean vector μ and covariance matrix Σ. The matrix Σ and its inverse, the precision matrix $K = \Sigma^{-1}$, are both symmetric and positive definite [30]. Note that we can also define the density in terms of K as $P(x \mid \mu, K)$ by replacing Σ with K^{-1} in (4.1). It is useful for us to focus on this alternative definition because pairwise independence statements as described in Section 4.2 are directly represented in the precision matrix K. Specifically, the pairwise independence statement $(x_i \perp x_j \mid x_{\overline{\{i,j\}}})$ is true if and only if $K_{ij} = 0$ (and, because K is symmetric, $K_{ji} = 0$). Therefore, every multivariate normal distribution implicitly incorporates a MRF graph structure in its precision matrix. This type of MRF model is called a GMRF because it represents a multivariate Gaussian distribution.

To understand the connection between a graph structure and the precision matrix of a multivariate normal distribution, it is necessary to know how the density $P(x \mid \mu, K)$ changes when the values of a subset of the variables are observed. Assume that the values of the variables x_B are observed and that the unobserved variables are x_A where $A = V - B$. Partitioning the variables in this manner leads to a corresponding partitioning of the parameters in the mean vector and the precision matrix:

$$\mu = \begin{bmatrix}\mu_A \\ \mu_B\end{bmatrix} \text{ and } K = \begin{bmatrix}K_{A,A} & K_{A,B} \\ K_{B,A} & K_{B,B}\end{bmatrix}.$$

The conditional density $P(x_A \mid x_B)$ is multivariate normal with parameters derived from the components of these partitionings [7]. Treating the observed variables x_B as constants in (4.1) yields the conditional mean

$$\mu_{A|B} = \mu_A - K_{A|B}^{-1} K_{A,B}(x_B - \mu_B) \tag{4.2}$$

and the conditional precision matrix $K_{A|B} = K_{A,A}$. It is also useful to note that the marginal density $P(x_A)$ is also multivariate normal and that we can determine its parameters by integrating the variables x_B out of (4.1) [7]. In this case, the mean is simply μ_A and the precision matrix is

$$K_A = K_{A,A} - K_{A,B} K_{B,B}^{-1} K_{B,A}.$$

The parameters of the marginal density are useful when we consider learning with EM in Section 4.5.1. With the parameter update equations for $\mu_{A|B}$ and $K_{A|B}$, we can establish the link between graph structures and precision matrices.

This connection is made in the proof of the following claim. Speed and Kiiveri prove a similar result for more general conditional independence statements but their proof is significantly more complex [43]. Here we focus on the relatively simple case of pairwise independence statements. Because the multivariate normal density $P(x \mid \mu, K) > 0$ for all x, the pairwise and local independence statements defined in Section 4.2 both hold for the graph structure encoded in the precision matrix K. This is important because PIA constructs a graph structure based on local independence statements derived from a set of KEGG pathways. It then invokes the pairwise independence statements about that graph to constrain the elements of K prior to learning. The following claim establishes the connection between K and the pairwise independence statements of a GMRF graph structure.

Claim 4.1. *If the variables $x_1, x_2, \ldots, x_p$ follow a multivariate normal probability distribution with precision matrix $K = [K_{ij}]$, then $K_{ij} = K_{ji} = 0$ if and only if $(x_i \perp x_j \mid x_{\overline{\{i,j\}}})$.*

Proof. First, assume that $K_{ij} = K_{ji} = 0$ for a multivariate normal density with precision matrix $K = [K_{ij}]$. Let $A = \{i, j\}$ and $B = V - \{i, j\}$ represent the indices of unobserved and observed variables respectively. For convenience, let $I = \{i\}$ and $J = \{j\}$. Based on the update equation, the precision matrix in the conditional density $P(x_A \mid x_B)$ has a simple diagonal form:

$$K_{A|B} = \begin{bmatrix} K_{ii} & K_{ij} \\ K_{ji} & K_{jj} \end{bmatrix} = \begin{bmatrix} K_{ii} & 0 \\ 0 & K_{jj} \end{bmatrix}.$$

Because K is positive definite, the values K_{ii} and K_{jj} must be non-zero. Thus, the inverse of this matrix, the conditional covariance matrix, is

$$K_{A|B}^{-1} = \Sigma_{A|B} = \begin{bmatrix} \frac{1}{K_{ii}} & 0 \\ 0 & \frac{1}{K_{jj}} \end{bmatrix}.$$

Consider the effect that this diagonal form has on the conditional mean:

$$\begin{aligned} \mu_{A|B} &= \mu_A - K_{A|B}^{-1} K_{A,B}(x_B - \mu_B) \\ &= \begin{bmatrix} \mu_i \\ \mu_j \end{bmatrix} - \begin{bmatrix} \frac{1}{K_{ii}} & 0 \\ 0 & \frac{1}{K_{jj}} \end{bmatrix} \begin{bmatrix} K_{I,B} \\ K_{J,B} \end{bmatrix} (x_B - \mu_B) \\ &= \begin{bmatrix} \mu_i - \frac{1}{K_{ii}} K_{I,B}(x_B - \mu_B) \\ \mu_j - \frac{1}{K_{jj}} K_{J,B}(x_B - \mu_B) \end{bmatrix} = \begin{bmatrix} \mu_{I|B} \\ \mu_{J|B} \end{bmatrix}. \end{aligned}$$

The initial assumption that $K_{ij} = K_{ji} = 0$ decouples the parameters of the conditional density corresponding to the mean values of x_i and x_j. Consequently, the conditional

density has the form

$$\begin{aligned}
P(x_A \mid x_B) &\propto \exp\left[-\frac{1}{2}(x_A - \mu_{A|B})^T K_{A|B}(x_A - \mu_{A|B})\right] \\
&= \exp\left[-\frac{1}{2}(x_i - \mu_{I|B})K_{ii}(x_i - \mu_{I|B}) - \frac{1}{2}(x_j - \mu_{J|B})K_{jj}(x_j - \mu_{J|B})\right] \\
&= \exp\left[-\frac{1}{2}(x_i - \mu_{I|B})K_{ii}(x_i - \mu_{I|B})\right] \exp\left[-\frac{1}{2}(x_j - \mu_{J|B})K_{jj}(x_j - \mu_{J|B})\right] \\
&\propto P(x_i \mid x_B)P(x_j \mid x_B)
\end{aligned}$$

implying that $(x_i \perp x_j \mid x_B)$. The other direction of the proof starting with the pairwise independence statement is similar. The conditional independence implies that the conditional covariance has a diagonal form. Inverting yields a diagonal precision matrix, thus the assumption implies $K_{ij} = K_{ji} = 0$. Therefore, the statement $(x_i \perp x_j \mid x_B)$ is true if and only if $K_{ij} = K_{ji} = 0$. □

The consequence of this result is that the graph structure of a GMRF corresponds to a pattern of zero-valued elements in the precision matrix of a multivariate normal distribution. This is based on the definition of the pairwise independencies of a graph structure given in Section 4.2. The PIA system exploits this correspondence to incorporate a graph derived from metabolic pathways into the representation of a multivariate normal distribution. If the pathways were unknown, then PIA could make no assumptions about conditional independencies. This corresponds to a complete graph where every pair of nodes are adjacent to each other. The complete graph has the maximum number of parameters possible associated with it. PIA uses the pathway-based graph structure to reduce the number of parameters in the models that it learns.

In particular, a multivariate normal density over p variables has $2p + \frac{p(p-1)}{2}$ total parameters. The mean vector μ accounts for p of these and there are p variances (the diagonal elements of Σ). Also, there are $\frac{p(p-1)}{2}$ possible ways to select distinct pairs of elements from a set of size p ignoring the order of elements. This is the number of unique covariance elements in Σ because this matrix is symmetric. In a GMRF model, where the zero pattern is known, the number of parameters is reduced because a certain number of elements of K are fixed. In this case, the number of parameters reduces to $2p + |E|$ because only the edges correspond to non-zero elements of the precision matrix. The zero pattern in the covariance matrix Σ, if any, is generally different from that in K but the number of parameters is effectively reduced.

Under the assumption that the data is is jointly normal, several standard machine learning models have interpretations as GMRF models. These models have varying numbers of parameters, some with less than PIA and some with more. We turn now to a discussion of these models.

4.4 GMRF Models

Different types of graphical model are determined by how the underlying graph structure is obtained. One general approach involves learning a graph structure from the same data set used to learn parameters. We discuss two specific algorithms for GMRF structure learning in Section 4.5. Another approach is to specify a fixed graph structure in advance of parameter learning. One way to accomplish this is to use knowledge provided by a domain expert to determine an appropriate structure. The metabolic pathways used by PIA constitute a form of expert knowledge. There are also common structures that a probability distribution of interest is assumed to follow. In general, these structures incorrectly represent

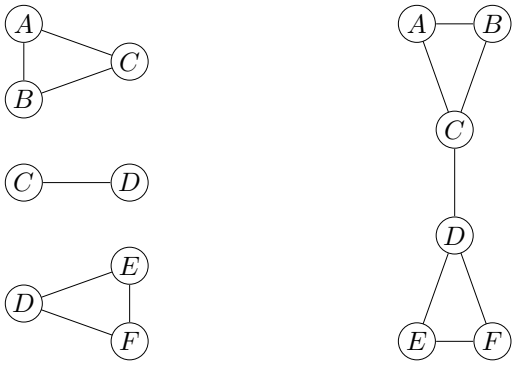

Figure 4.2: Graph structure representation of three generic reactions separately (left) and their combined (right) representation employed by PIA.

the independence properties of the distribution but they tend to produce good classifiers in practice.

4.4.1 Knowledge-based Models

The basis of the PIA system is a graph structure derived from a set of metabolic pathways. PIA treats these pathways as a single large set of chemical reaction equations. Of the several possible graph constructions discussed in Section 3.2, we employ the WF construction in PIA. This construction joins all pairs of metabolites in a reaction with edges, thus the set of nodes corresponding to each reaction forms a clique. A combined graph representing multiple reactions is constructed in a natural way. The set of nodes in the combined graph is the union of the nodes in the graph corresponding to each reaction. A node is uniquely determined by the metabolite that it represents. An edge is included in the combined graph that joins a pair of metabolite nodes if and only if those metabolites participate in a reaction together. As an example, consider the generic chemical reaction equations

1. $A + B \rightarrow C$
2. $C \leftrightarrow D$
3. $D \leftrightarrow E + F$

and their separate and combined graph representations shown in Figure 4.2.

The WF graph construction leads to a natural interpretation of the combined graph in terms of probabilistic independence statements about the distribution of metabolic profile measurements. To understand this interpretation, consider a metabolite v. Metabolic pathways describe how v is produced and consumed by the chemical reactions that involve it. The only metabolites that can directly influence the concentration of v are the other metabolites that participate in these reactions. Given the concentrations of these metabolites, the concentration of v is independent of every other metabolite. This is a local conditional independence statement about the concentration of v. The WF construction produces a graph structure that captures precisely these conditional independence statements about the concentrations of metabolites in a set of metabolic pathways.

4.4.2 Heuristic Models

One standard model with very few parameters is the NB model. This model is based on the assumption that metabolite concentrations are completely independent of each other, given

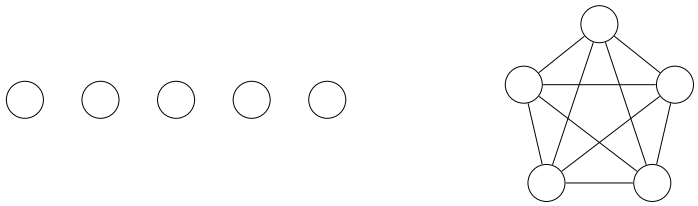

Figure 4.3: Example graph structures for distributions with five variables used by the Naïve Bayes model (left) and Full dependence model (right).

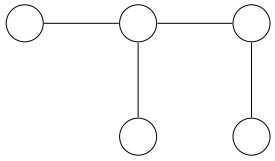

Figure 4.4: Example tree-structured graph learned by Tree-augmented Naïve Bayes.

the class of the patient. In this context, the assumption is clearly false but the model tends to produce classifiers that perform well in practice [17]. When accurate classification is the only goal, correctly representing the class-conditional densities is not necessary. In the NB model, the class-conditional probabilities are represented by graph structures containing no edges. This model contains the smallest possible number of parameters for a multivariate normal distribution.

The Full model is a related model that assumes the concentration of each metabolite depends on every other metabolite. In this model, every pair of nodes is joined by an edge into a single large clique. Unlike the NB model, the Full model makes no independence statements about the distribution it represents. It has the maximum number of parameters possible under the assumption that the data follows a multivariate normal distribution. Example graph structures for the NB and Full models with five variables are shown in Figure 4.3.

Slightly more sophisticated models are obtained by relaxing the assumptions of the NB model. Two examples are TAN and the related MTAN models [20]. These involve learning graph structures from the data with the restriction that the learned graph has a tree structure. An example tree-structured graph is given in Figure 4.4. The difference between TAN and MTAN is that both class-conditional distributions in TAN share the same structure but in MTAN they are allowed to differ. TAN uses all instances in the data set to learn a single structure for both distributions. In MTAN, the training set is split into positive and negative instances and the two class-conditional GMRF structures are learned from these two sets separately. Details of this structure learning algorithm are described in the next section.

4.5 GMRF Learning Algorithms

This section discusses the two dimensions of GMRF learning introduced above. From the perspective of PIA, the most important of these is parameter learning. In this problem, there is a previously known graph structure, such as one derived from metabolic pathway knowledge. Given this structure, we want to compute the mean μ and precision matrix K consistent with the constraints implied by the structure. We are particularly interested in

the precision matrix K because the mean μ is not constrained by the structure. Most of the models we consider have a structure specified in advance but there are techniques for learning the structure itself from a data set when one is not provided. In this section, we discuss algorithms for structure learning, parameter learning and one that simultaneously performs both.

4.5.1 Learning Parameters

This thesis employs two GMRF parameter learning algorithms. The first of these is an unconstrained approach that finds the most likely parameter values given a data set of examples. The other algorithm is based on this principle but it also incorporates constraints on the learned precision matrix K determined by an underlying graph structure. When learning the parameters for models with latent variables, we incorporate these two algorithms into the EM algorithm introduced in Section 2.4.2. These algorithms work as follows.

- **Maximum Likelihood (ML)** is a general technique for estimating the parameters of a model from data. Bishop describes it in the context of data following a multivariate normal distribution [7]. The approach assumes that all of the instances $x_r \in \mathcal{D}$ are independent of each other. In this case, we can easily express the probability of the instances in the data set given a set of parameters μ and Σ. The logarithm of this expression is usually taken because the result is easier to maximize. The resulting expression will have the same maximum because the log function is monotonic. Specifically, this expression is

$$\log P(\mathcal{D} \mid \mu, \Sigma) = \sum_{r=1}^{n} \log P(x_r \mid \mu, \Sigma) \tag{4.3}$$

  where the probability of a single instance is given by (4.1). Typically, to obtain the maximum likelihood parameter settings μ and Σ, we differentiate with respect to these variables and set the resulting expressions to zero. For a multivariate normal distribution, manipulation of these expressions yields the standard estimates

$$\hat{\mu} = \frac{1}{n} \sum_{r=1}^{n} x_r$$
$$\hat{\Sigma}' = \frac{1}{n} \sum_{r=1}^{n} (x_r - \hat{\mu})(x_r - \hat{\mu})^T.$$

  It turns out that the ML estimate $\hat{\Sigma}'$ is biased because its expected value is not equal to Σ, the true covariance matrix. This problem is easy to correct however, thus we use the unbiased estimate of Σ instead:

$$\hat{\Sigma} = \frac{1}{n-1} \sum_{r=1}^{n} (x_r - \hat{\mu})(x_r - \hat{\mu})^T. \tag{4.4}$$

  The estimate $\hat{\Sigma}$ is problematic for PIA because it does not account for graph structure constraints on the precision matrix. We can use $\hat{\Sigma}$ for the NB and Full models and $\hat{\mu}$ for all models but a different algorithm is needed to incorporate the constraints imposed by the GMRF graph structures of TAN, MTAN and PIA.

- **Speed and Kiiveri (SK)** describe an algorithm for maximum likelihood parameter estimation that incorporates constraints on the precision matrix given by a GMRF graph structure [43]. Given a positive definite covariance matrix estimate $\hat{\Sigma}$ and a

known graph structure $G = (V, E)$ for the GMRF, the SK algorithm produces a precision matrix estimate $\hat{K}$ such that

$$(\hat{K}^{-1})_{ij} = \hat{\Sigma}_{ij} \text{ if } \{i, j\} \in E \text{ or } i = j$$
$$\hat{K}_{ij} = 0 \text{ if } \{i, j\} \notin E.$$

It does this by performing a series of matrix updates over elements of $\hat{K}$ corresponding to cliques in the graph G. These updates are repeated until convergence to a fixed matrix $\hat{K}$. Speed and Kiiveri prove that there is one such matrix and that their algorithm computes it efficiently.

If the SK algorithm updates are performed over maximal cliques, then the algorithm executes faster than if they are not. Although computing the maximal cliques of a graph is computationally inefficient in the worst case, we pursue an approach based on maximal clique updates for two reasons. First, the metabolic pathway graphs derived from KEGG are relatively small. Also, the structure of the metabolic network for well-studied organisms does not change frequently. Thus, we invest the computational effort necessary to compute the maximal cliques of the metabolic pathway graph structures of interest and store the results for repeated use in the future. Section 5.3.2 describes how PIA obtains the maximal cliques of a graph.

- **Expectation Maximization (EM)** is a general approach for handling missing values when performing maximum likelihood parameter estimation [16]. Section 2.4 discussed EM in general terms and here we describe how PIA uses EM. PIA employs EM to estimate the parameters of GMRF models with latent variables. Let A and B represent the indices of unobserved and observed variables respectively. Our approach begins with a random mean vector $\hat{\mu}$ and precision matrix $\hat{K}$ with appropriate elements set to zero to incorporate the constraints from the GMRF graph structure. For each instance $x \in \mathcal{D}$, the expectation step must compute the mean of the conditional distribution $P(x_A \mid x_B)$ to fill in the missing values. To do this, we set $x_A = \hat{\mu}_{A|B}$ where $\hat{\mu}_{A|B}$ is defined according to (4.2). After filling in all missing values in this manner, PIA computes a covariance estimate $\hat{\Sigma}$ for the SK algorithm. The maximization step uses the ML and SK algorithms to learn a new mean vector $\hat{\mu}$ and precision matrix $\hat{K}$ respectively. The two steps of EM are repeated until convergence to a fixed $\hat{\mu}$ and $\hat{K}$.

4.5.2 Learning Structure

We consider two different algorithms for GMRF structure learning. They are very different in the types of constraints they place on the learned structures. In the limit of infinite data, the ML approach above would produce the correct structure [4]. Our data sets are relatively small, thus some constraints are necessary to produce a good structure that adequately represents the observed distribution of the data. The following algorithms represent different ideas about how to impose these constraints.

- **Tree-augmented Naïve Bayes (TAN)** and the related MTAN are models that generalize NB to account for certain dependencies among the variables [20]. These models allow for a restricted set of edges among the nodes in a graph underlying a probability distribution. Specifically, a tree structure is imposed on this graph. TAN learns a tree structure from a data set with the property that no other tree structure could assign a larger likelihood to the data [20]. TAN was originally described in the context of directed graphical models, called Bayesian Networks, with discrete variables. The ideas also apply to undirected GMRF models of continuous variables, thus we describe TAN graph learning in this context.

TAN structure learning is based on the idea of conditional mutual information between two variables [14]. When the class variable is binary, the conditional mutual information

$$I(x_i, x_j \mid c) = \sum_{c \in \{+,-\}} \int\int P(x_i, x_j \mid c) P(c) \log \frac{P(x_i, x_j \mid c)}{P(x_i \mid c) P(x_j \mid c)} \, dx_i \, dx_j$$

measures the dependence between two variables x_i and x_j given the class c. The distributions $P(c)$, $P(x_i \mid c)$, $P(x_j \mid c)$ and $P(x_i, x_j \mid c)$ are estimated from the instances in the data set. If these instances are assumed to follow a joint normal distribution the computation of $I(x_i, x_j \mid c)$ simplifies to

$$I(x_i, x_j \mid c) = \frac{n^+}{n} \frac{1}{2} \log \left(\frac{\hat{\Sigma}^+_{ii} \hat{\Sigma}^+_{jj}}{\det(\hat{\Sigma}^+)} \right) + \frac{n^-}{n} \frac{1}{2} \log \left(\frac{\hat{\Sigma}^-_{ii} \hat{\Sigma}^-_{jj}}{\det(\hat{\Sigma}^-)} \right) \tag{4.5}$$

where $\hat{\Sigma}^+$ and $\hat{\Sigma}^-$ are covariance matrix estimates for the marginal distributions $P(x_i, x_j \mid c = +)$ and $P(x_i, x_j \mid c = -)$ respectively [3]. After first computing the value $I(x_i, x_j \mid c)$ for every pair of variables x_i and x_j in the data set, TAN then constructs a complete graph where every pair of nodes is joined via an edge. Each edge in this graph $\{i, j\}$ is assigned a weight

$$w(\{i, j\}) = I(x_i, x_j \mid c).$$

This weight is the conditional mutual information between the corresponding variables. Next, TAN produces the learned graph structure by computing a maximum spanning tree over this weighted graph [13]. For Bayesian networks with discrete variables, TAN would then direct the edges in this tree from a selected root node and learn local parameters for each node. For GMRF models, we keep the undirected tree and apply the SK algorithm to learn the parameters.

As described, TAN learns a single graph structure over an entire data set to represent the two class-conditional distributions. MTAN is an alternative approach that allows these distributions to have different structures depending on the class value c. The approach is the same as that of Chow and Liu [12] and works as follows. The same idea as regular TAN is used except that the training data set is divided into two data sets according to the class. Each of these data sets is used to learn the structure for the corresponding class. Again, SK is used to learn GMRF model parameters given each of these learned structures.

- **Graphical Lasso (GL)** is a learning algorithm that maximizes a penalized log-likelihood equation to estimate the precision matrix K from a data set [19]. Specifically, GL maximizes

$$\log P(\mathcal{D} \mid \mu, K) - \rho \|K\|_1 \tag{4.6}$$

where the penalty $\|K\|_1 = \sum_i \sum_j |K_{ij}|$ is the standard L_1 norm. This penalty has the property that it tends to force many of the K_{ij} values to zero. Note that any value $K_{ij} = 0$ reduces the value of the log-likelihood function but it also effectively eliminates a parameter from the model. Only the most important of these parameters, equivalent to graph edges in a GMRF model, are allowed into the model. Including such a penalty is useful for structure learning from a finite data set because the ML solution is unlikely to produce a precision matrix K with elements that are exactly equal to zero [4].

The value of ρ influences the exact number of edges in the resulting graph. As ρ increases, more of the K_{ij} become zero producing a graph with fewer edges. It is

possible to generalize the scalar ρ into a matrix so that every possible edge $\{i, j\}$ has an associated value ρ_{ij}. This allows us to incorporate knowledge about the underlying graph structure into the GL algorithm. Given a graph $G = (V, E)$, we set ρ_{ij} to a very large penalty for the non-edges ($\{i, j\} \notin E$) forcing the corresponding $K_{ij} = 0$. For the edges $\{i, j\} \in E$, ρ_{ij} is set normally allowing the GL algorithm to determine what edges from the graph to keep. Thus, we can use GL to learn a structure constrained by prior knowledge from KEGG.

The penalized log-likelihood equation (4.6) is convex in K and there are a variety of efficient algorithms to optimize it [19]. The GL algorithm is one such algorithm that is based on a form of penalized linear regression called Lasso. GL sequentially performs Lasso regression to predict one variable given the others in the data set. Appropriately combining the solutions to these regression problems produces an estimate $\hat{K}$ that maximizes (4.6). Iterating the procedure and updating this estimate at each step converges to the optimal estimate $\hat{K}$. In practice, the GL algorithm is significantly faster than the other algorithms for this problem.

4.6 GMRF Classifiers

At this point, we have the tools necessary to obtain a GMRF graph structure $G = (V, E)$ as well as the corresponding parameter values μ and K. We use these GMRF models to represent joint probability densities over metabolic profile measurements. Specifically, the models represent the class-conditional densities $P(x_r \mid c_r = +)$ and $P(x_r \mid c_r = -)$ corresponding to metabolic profiles of diseased patients and healthy patients respectively. Naturally, the best classification we can make is to predict the most probable class given a metabolic profile or, more precisely, the value $c_r \in \{+, -\}$ that maximizes $P(c_r \mid x_r)$. To compute this value, we employ Bayes' rule

$$P(c_r \mid x_r) = \frac{P(x_r \mid c_r)P(c_r)}{P(x_r)}$$

where the class-conditional densities corresponds to the likelihood term $P(x_r \mid c_r)$. The computation entails selecting

$$c = \arg\max_{c_r} P(c_r \mid x_r) = \arg\max_{c_r} \frac{P(x_r \mid c_r)P(c_r)}{P(x_r)} = \arg\max_{c_r} P(x_r \mid c_r)P(c_r)$$

where the denominator term $P(x_r)$ is dropped because it does not depend on c_r. In other words, we select the class with the largest posterior probability. Since we can represent the likelihood, it is only necessary to compute the class prior probabilities for each class. This is easily accomplished by simply counting the patients from each class in the training set. The result is a Bayesian classifier based on GMRF models.

To make a prediction about the class of an undiagnosed patient q, we must evaluate the class-conditional likelihood densities. Consider the case where we assume the patient is positive, thus $c_q = +$. If the model has no latent variables (and the metabolic profile x_q contains no missing values), then evaluating the likelihood involves inserting the values x_q into the multivariate normal density of (4.1) determined by the GMRF parameters μ^+ and K^+. In the case that there are latent variables or missing values in x_q, we instead compute the parameters of the marginal density over the observed values as described in Section 4.3. We then use these parameters to evaluate the marginal density of the observed values in x_q. The same procedure is used to evaluate the likelihood of x_q assuming that the patient q is negative. The two likelihood values are used to determine the most likely class for the patient.

Chapter 5

Methods

This chapter describes how to combine the ideas developed in previous chapters to derive the PIA system. In particular, it discusses how PIA converts a set of metabolic pathways from KEGG into a GMRF graph structure. These structures are far too large to apply learning algorithms directly. Thus, we develop some techniques for simplifying them to make parameter learning feasible. Additional ideas that are important to PIA are described here as well. These ideas include a technique for computing a covariance matrix estimate from a small number of samples and an algorithm to compute the maximal cliques of a graph to improve the performance of the SK algorithm.

5.1 Metabolic Graph Construction

At this point, we have established the motivation for treating a set of metabolic pathways as a graph structure and discussed specific approaches for generating these graphs. We described the interpretation of these graphs in terms of probabilistic independence statements and their use in representing probability distributions with GMRF models. PIA combines these ideas to create a pathway-based disease classification tool. The PIA system employs the WF graph construction discussed in Section 3.2. This and other representations were described there in terms of individual reactions. An individual pathway consists of several reactions however. PIA generates a combined graph representation of a set of pathways as described in Section 4.4.

A metabolic pathway in KEGG consists of a set of chemical reaction equations. For a given pathway, a specific organism will have a subset of these reactions corresponding to the enzymes encoded in its genome. The details of how KEGG represents this information are contained in Appendix A. PIA generates a graph representation of a given set of pathways as follows. For every reaction that is on one or more of the pathways, it generates a graph representation of the reaction following the WF construction as described in Section 3.2. The nodes in this graph are identified by the name of the corresponding metabolite. Given this set of reaction graphs, PIA generates a combined graph over all nodes and edges in the reaction graphs. In this combined graph, nodes are uniquely determined by their metabolite names. Thus, a metabolite that occurs in many reactions will only have one node in this graph. An example of this procedure applied to several generic reaction equations is shown in Figure 4.2.

The combined graph that results from using the WF representation has an intuitively appealing interpretation in terms of probabilistic independence statements. Recall that in this representation, a pair of metabolites are connected via an edge if and only they participate in a reaction together. According to the independence properties of an undirected graph described in Section 4.2, we can interpret the resulting graph structure in a specific

Pathway ID	Pathway Name
00010	Glycolysis / Gluconeogenesis
00020	Citrate cycle (TCA cycle)
00220	Urea cycle and metabolism of amino groups
00251	Glutamate metabolism
00252	Alanine and aspartate metabolism
00260	Glycine, serine and threonine metabolism
00271	Methionine metabolism
00272	Cysteine metabolism
00280	Valine, leucine and isoleucine degradation
00290	Valine, leucine and isoleucine biosynthesis
00300	Lysine biosynthesis
00310	Lysine degradation
00330	Arginine and proline metabolism
00340	Histidine metabolism
00350	Tyrosine metabolism
00360	Phenylalanine metabolism
00380	Tryptophan metabolism
00400	Phenylalanine, tyrosine and tryptophan biosynthesis
00410	beta-Alanine metabolism
00430	Taurine and hypotaurine metabolism
00440	Aminophosphonate metabolism
00450	Selenoamino acid metabolism
00460	Cyanoamino acid metabolism
00471	D-Glutamine and D-glutamate metabolism
00472	D-Arginine and D-ornithine metabolism
00473	D-Alanine metabolism
00480	Glutathione metabolism

Table 5.1: Metabolic pathways from KEGG that are relevant to the metabolism of cachexia.

way. In particular, we invoke the local independence properties of an undirected graphical model. In this context, the nodes are taken to represent random variables corresponding to metabolite concentrations. The graph implies that the concentration of a metabolite is independent of every other metabolite given the concentrations of those metabolites that are participants in a common reaction. Because we are only modelling the metabolite concentrations and their relationships, nothing else in the model could have a direct influence on the concentration of a metabolite.

To simplify the default graph structure, we use the idea of currency metabolites from Section 3.3. Because we use the WF graph construction, we also use the corresponding currency metabolite list that is given in Table 3.1. Currency metabolites are not included in the graph structures that PIA uses. Their exclusion takes place during the reaction graph construction step. Any metabolite on the list of currency metabolites is treated as if it did not participate in any reactions even if it does according to KEGG. KEGG generally does not represent complete reaction equations in its metabolic pathways however. Instead, only the primary metabolites of a reaction are included, thus the removal of currency metabolites does not have a major impact on the resulting graph structures. This is because the currency metabolites typically are not the most important metabolites in a reaction. They are defined as those metabolites that are available in sufficient quantity that they do not affect the operation of the reactions where they participate. Thus, the concentrations of non-currency metabolites do not depend on the currency metabolites. In the context of probabilistic graphical models, this constitutes sufficient justification for ignoring them.

	reactions	nodes	edges	parameters
H. sapiens (all pathways)	1,651	1,504	1,795	4,803
currency metabolites removed		1,493	1,731	4,717
H. sapiens (cachexia pathways)	417	391	506	1,288
currency metabolites removed		386	487	1,259
C. elegans	918	1,004	1,108	3,116
currency metabolites removed		994	1,057	3,045
S. scrofa	897	1,022	1,022	3,066
currency metabolites removed		1,012	995	3,019

Table 5.2: Metabolic graph structure properties, before and after removing currency metabolites.

Another way to control the size of the graph structures is through the sets of metabolic pathways used to generate them. We consider two different sets of metabolic pathways from KEGG. The first set consists of all 165 pathways that KEGG designates as metabolic pathways. We also consider using a set of expert-determined pathways that include only the pathways that are most relevant to the disease mechanism of cachexia. Activity in these pathways is expected to differ in cachexic patients when compared to patients without the disease. The list of these 27 cachexia pathways is given in Table 5.1. This list shows that PIA can incorporate additional biological knowledge through the selection of pathways it uses to derive its GMRF graph structure.

The PIA system is applied to three data sets, each containing metabolic profiles from a different organism. Properties of the default graph structures for these three organisms are given in Table 5.2. The table suggests that the models corresponding to these graphs have many parameters to learn. Recall that the number of parameters in a GMRF model based on graph $G = (V, E)$ is $2|V| + |E|$. Our data sets have around 100 instances each, not nearly enough to accurately estimate over 1,000 parameters as required for the graphs in Table 5.2. The table shows that removing currency metabolites does not significantly reduce the number of parameters. This is because KEGG already ignores most currency metabolites on its pathways.

As indicated in Table 5.2, the default graph structure resulting from the WF construction procedure applied to all human metabolic pathways contains 1,493 nodes and 1,731 edges. Only 48 of the nodes in this graph correspond to measured metabolites in the cachexia data set. The remaining 1,445 nodes correspond to latent variables. Clearly, a metabolic profile only measures a small portion of the compounds that are involved in the human metabolic network. Since this number of latent variables is far too large to apply EM directly, we consider some ideas to further simplify the graph structure. These are intended primarily to remove latent variables but they also help to significantly reduce the number of parameters in the models used by PIA.

5.2 Metabolic Graph Transformations

This section introduces two graph operations that PIA employs to reduce the size and complexity of its GMRF graph structures. Assume that there is a graph structure $G = (V, E)$ constructed from a set of metabolic pathways according to the WF procedure. For a given data set, the set of nodes V is implicitly partitioned into two subsets M and L corresponding to measured and latent (unmeasured) variables respectively. As noted above, our cachexia data set only covers 48 of the 1,445 metabolites in the human metabolic network. The other data sets described in Chapter 6 poorly cover the networks of their respective organisms as well. Thus, the default graphs contain too many latent variables to effectively learn

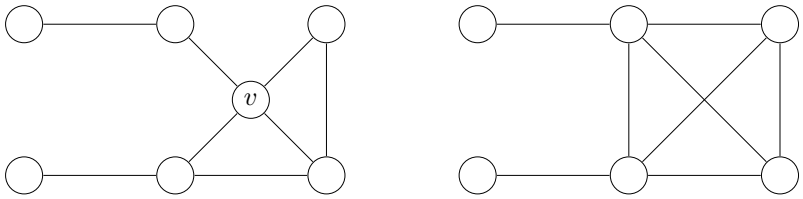

Figure 5.1: Example of the marginalize operation applied to a node v of a graph.

1: **procedure** MARGINALIZE(G, v)
Require: $G = (V, E)$ is an undirected graph with $v \in V$
2: $E \leftarrow E \cup \{\{i, j\} \mid i, j \in N(v) \text{ and } i \neq j\}$ ▷ connect pairs of neighboring nodes
3: $E \leftarrow E - \{\{i, v\} \mid i \in N(v)\}$ ▷ remove edges incident to v
4: $V \leftarrow V - \{v\}$ ▷ remove the node v
5: **end procedure**

Algorithm 5.1: Marginalize a variable out of a distribution modeled as a GMRF.

the parameters of the corresponding models. The two transformations described here are intended to reduce the number of latent variables in a graph while attempting to preserve its independence properties.

5.2.1 Marginalize

Marginalization is the process of integrating a variable out of a distribution to produce the distribution over the remaining variables [30]. The conditional independence statements satisfied by the variables in the resulting marginal distribution can be derived from those of the original distribution. The marginalize operation implements this derivation. Given a GMRF graph structure and a variable, the marginalization process produces a graph for the marginal distribution over the remaining variables. The operation is implemented as a simple graph transformation.

Typically, the marginalize operation will introduce some new dependencies into the marginal distribution. The fact that a variable is not modeled in a distribution does not mean it no longer exists. A marginalized-out variable still influences and is influenced by other the variables in the marginal distribution. In addition to removing a variable node, the marginalize operation accounts for these influences in the graph it produces. Algorithm 5.1 defines the graph transformation implemented by the marginalize operation.

The transformation considers a node v and its neighborhood $N(v)$. Every pair of nodes in $N(v)$ must be joined via an edge producing a clique over them. The conditional independence of any two variables with nodes in $N(v)$ always requires conditioning on the variable x_v according to the three characterizations given in Section 4.2. This is no longer possible in the marginal distribution however. Without x_v, the possible dependencies can only be accounted for by the edges added on line 2. The remainder of the marginalize operation removes edges incident to v and the node v itself. An example of applying the marginalize operation to a node in a graph is shown in Figure 5.1.

The marginalize operation introduces the minimum number of dependencies to avoid introducing independencies that did not exist in the original graph. To see that the marginalize operation adds only the necessary edges and no more, first consider the possibility that it adds an edge unnecessarily. Suppose that the graph $\overline{G} = (\overline{V}, \overline{E})$ is the result of marginalizing $v \in V$ from the graph $G = (V, E)$ and that the edge $\{a, b\}$ was unnecessarily added

		nodes	edges	parameters
all pathways	default graph	1,493	1,731	4,717
	fully marginalized	48	572	668
	greedily marginalized	80	259	419
	merged	55	65	175
	merged + greedily marginalized	49	59	157
cachexia pathways	default graph	386	487	1,259
	fully marginalized	35	284	354
	greedily marginalized	47	120	214
	merged	42	52	136
	merged + greedily marginalized	36	43	115

Table 5.3: Transformed metabolic graph structure properties for human graphs after removing currency metabolites.

by the algorithm. Now assume that $\{a, b\} \notin \overline{E}$ and consider the implications. In this case, $a, b \in N(v)$ because marginalizing only joins pairs of nodes in $N(v)$ as seen on line 2 of Algorithm 5.1. Also, because the algorithm made the decision to join a and b, $\{a, b\} \notin E$ is true. The set of nodes $C = V - \{a, b\}$ must separate a and b in $\overline{G}$. The assumption $\{a, b\} \notin \overline{E}$ implies that any path between a and b must pass through some node in C. The set C does not separate a and b in G however because of the path a, v, b. Thus, $\overline{G}$ implies that $(x_a \perp x_b \mid x_C)$ but G does not. This represents a newly introduced conditional independence, contradicting the assumption that the edge $\{a, b\}$ was unnecessarily added. Therefore, all edges added by the marginalize operation are necessary.

The marginalize operation also does not introduce any independencies by failing to add a required edge. To see this, assume that $\overline{E}$ should contain $\{a, b\}$ but it does not, thus $\{a, b\} \notin \overline{E}$. If $\{a, b\} \in E$, then the marginalize operation would not have removed it because the operation only removes edges incident to v and both a and b are different from v. Thus, $\{a, b\} \notin E$ must hold for consistency with the assumption. The assumption also would be false if $a, b \in N(v)$ so at least one of a, b is not in $N(v)$. Assume without loss of generality that $a \notin N(v)$. The set of nodes $C = V - \{a, b\}$ must separate a and b in G. The only nodes not in C are a, b and v. The separation is true because any path from a to b through v must pass through a node in $N(v)$ that is different from b but this node is in C. Thus, the graph G implies $(x_a \perp x_b \mid x_C)$ but the assumption implies that $(x_a \perp x_b \mid x_C)$ is false. This contradiction means that the operation did not fail to add an edge to $\overline{G}$. Therefore, exactly the right set of edges is added by the marginalize transformation.

Applying this transformation to a node v of a graph affects the number of parameters in the corresponding GMRF model. The change can increase or decrease the number of parameters depending on how highly connected the neighborhood of v is. The operation always eliminates two parameters: one for the mean of x_v and the other for the corresponding diagonal precision matrix element. For node v, there exists some number k_v of edges among the nodes in $N(v)$. The maximum number of possible edges among $N(v)$ is $\frac{1}{2}|N(v)|(|N(v)|-1)$. Thus, the total change in the number of parameters is

$$-2 - \deg(v) + \frac{1}{2}|N(v)|(|N(v)| - 1) - k_v.$$

This number can be positive or negative depending on the connectivity of the nodes in $N(v)$. The change in the number of parameters for the example graph in Figure 5.1 is $-2 - 4 + \frac{1}{2} \times 4 \times 3 - 2 = -2$.

PIA uses the marginalize operation in two ways. One approach applies it to every latent variable node in the model. We call this approach full marginalization because it does not consider the effect on the number of parameters in the corresponding GMRF model. The

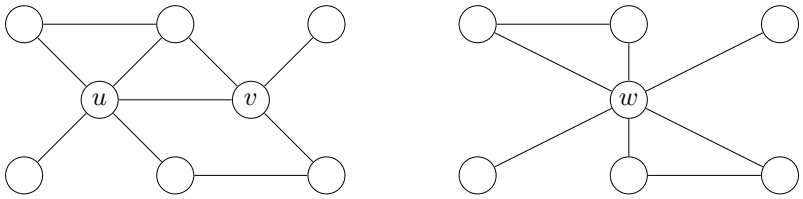

Figure 5.2: Example of a merge operation applied to the nodes u and v of a graph to produce a new node w.

1: **procedure** MERGE(G, u, v)
Require: $G = (V, E)$ is an undirected graph with $u, v \in V$ and $\{u, v\} \in E$
Ensure: $w \notin V$
2: $V \leftarrow V \cup \{w\}$ ▷ add a new node
3: $E \leftarrow E \cup \{\{i, w\} \mid i \in N(u) \cup N(v)\}$ ▷ connect new node to neighbors of u and v
4: $E \leftarrow E - \{\{i, u\} \mid i \in N(u)\}$ ▷ remove edges incident to u
5: $E \leftarrow E - \{\{i, v\} \mid i \in N(v)\}$ ▷ remove edges incident to v
6: $V \leftarrow V - \{u, v\}$ ▷ remove the nodes u and v
7: **end procedure**

Algorithm 5.2: Merge a pair of latent variables in a distribution modeled as a GMRF.

same graph is produced regardless of the order of marginalization (see Theorem C.1). The other approach takes the parameterization of the resulting models into consideration. This approach is called greedy marginalization because it marginalizes a latent variable node only if the transformed graph structure has fewer parameters than the original one. This process is repeated as long as such a node can be found, selecting the node that produces the greatest reduction in the number of parameters. Because marginalizing one node u can change the effect marginalizing v has on this number, we apply the operations in this greedy fashion. The effect that these two graph transformations have on the default graph structure are indicated in the "fully marginalized" and "greedily marginalized" rows of Table 5.3.

5.2.2 Merge

The merge operation operates on two nodes $u, v \in L$ at a time. The nodes u and v must be adjacent, thus $\{u, v\} \in E$. It removes the two nodes and replaces them with a single new node connected to all nodes to which either u or v was adjacent. The motivation for this operation stems from the typical arrangement of metabolites on a KEGG pathway. Many of these involve one compound being transformed into another and so on, forming a long path of latent variables in the default GMRF structures. Instead of modelling the effects of many latent variables on each other, we are only interested in the effects that they have on the observed variables. In contrast to the marginalize operation, the merge operation reduces the complexity of the GMRF model but it also preserves some latent variables to account for these effects. The merge operation that we devised is defined in Algorithm 5.2. An example of a merge operation performed on two adjacent nodes of a graph is shown in Figure 5.2.

The merge operation creates a new latent variable x_w and a node w for it that was not previously in V. This node will take the place of the nodes u and v that are merged. Merging is accomplished by joining every node in $N(u) \cup N(v)$ to the new node w. The nodes u and v and their incident edges are all removed from the graph. Because the node w did not exist and u and v both correspond to latent variables, w must also correspond to

a latent variable.

The merge operation preserves the independencies among variables corresponding to unaffected nodes without adding additional independencies among them. Because the operation is only applied to latent variable nodes, these are the only independencies that are important in the GMRF models. As described in Section 4.6, PIA uses marginal densities over the measured variables to compute the likelihood values needed to produce a classification. To understand the consistency between the initial and transformed graphs, consider the merge operation applied to adjacent nodes u and v of a graph $G = (V, E)$. Suppose that $\overline{G} = (\overline{V}, \overline{E})$ is the result of this transformation. Let A, B and C denote disjoint subsets of V such that neither A or B contains u or v (but C might).

If $(x_A \perp x_B \mid x_C)$ holds according to G then a similar statement $(x_A \perp x_B \mid x_{\overline{C}})$ is true according to $\overline{G}$ where $\overline{C}$ is defined as

$$\overline{C} = \begin{cases} C & \text{if } u \notin C \text{ and } v \notin C \\ C \cup \{w\} & \text{if } u \in C \text{ or } v \in C. \end{cases}$$

Consider a path $a, v_1, v_2, \ldots, v_k, b$ in G from $a \in A$ to $b \in B$ that is blocked by a node $c \in C$. We can translate this into a path in $\overline{G}$ by replacing occurrences of u and v with their merged counterpart w. If the node $c = u$ or $c = v$, then $w \in \overline{C}$ and this node blocks the path in $\overline{G}$. If c is not u or v, then by construction $c \in \overline{C}$ and c blocks the translated path because this node was not changed by the translation. Therefore, independencies among the nodes that the merge operation is not applied to are preserved in the resulting graph.

The merge operation should not introduce new independencies among these nodes either. To see why, consider that $(x_A \perp x_B \mid x_{\overline{C}})$ is true according to $\overline{G}$ and define the set of nodes C as

$$C = \begin{cases} \overline{C} & \text{if } w \notin \overline{C} \\ \overline{C} \cup \{u, v\} & \text{if } w \in \overline{C}. \end{cases}$$

Then the graph G implies that $(x_A \perp x_B \mid x_C)$ is true. As above, let $a, v_1, v_2, \ldots, v_k, b$ be a path in $\overline{G}$ from $a \in A$ to $b \in B$ that is blocked by a node $\overline{c} \in \overline{C}$. Translate the path into G by replacing w with u, v or the edge $\{u, v\}$ as needed depending on what nodes occur before and after w in the path to make it valid in G. If the node $\overline{c} = w$ then the replacement node(s) block the translated path in G. If $\overline{c}$ is not w then $\overline{c} \in \overline{C}$ and it also blocks the translated path because the node was not changed. Therefore, the merge operation does not introduce new independencies among the nodes it is not applied to either.

Like the marginalize operation, applying the merge operation to a graph changes the number of parameters in the corresponding GMRF model. In this case, however, the change is strictly a reduction in the number of parameters. Two nodes u and v are reduced to the one node w for a net decrease of two parameters. Four parameters are lost from u and v but two more are added for w. We also have that $\deg(u) + \deg(v) \geq \deg(w)$. This is because w is adjacent only to nodes in $N(u) \cup N(v)$ but there could be overlap. In other words, if $N(u) \cap N(v) \neq \emptyset$ then $\deg(u) + \deg(v) > \deg(w)$. Therefore, the total reduction in number of parameters is

$$2 + |N(u) \cap N(v)|.$$

The change in number of parameters for the graph in Figure 5.2 is $-2 - 1 = -3$.

The PIA system uses the merge operation to merge every pair of adjacent latent variables in a graph structure. The merge operation is repeatedly applied until no such pair exists. Note that this includes the w nodes in line 2 of Algorithm 5.2 because these are also latent variable nodes. As with the marginalize operation, the order of application of the merge operation is not relevant. The same graph is produced as a result (see Theorem C.2). The result of applying the merge operation in this manner is that some latent variables will remain in the structure. The nodes corresponding to these variables have the property that they are only adjacent to measured variable nodes.

After repeated application of the merge operation as described, a common pattern in the resulting metabolic graphs is to have a sequence of nodes u, v and w where u, v, w is a path, u is not adjacent to w, $v \in L$ and $u, w \in M$. In this case, the latent variable node v can be marginalized to produce a graph with strictly fewer parameters. Thus, the distribution modeled by the graph has fewer latent variable nodes as well. We add these marginalization steps after the merge operations are applied to obtain these simplified graphs. We expect that learning parameters for the resulting models will be easier.

As with the marginalization, applying merge to a graph structure significantly affects the number of parameters in the corresponding GMRF model. These effects are quantified in Table 5.3.

5.3 Other Issues

We describe two additional components of the PIA system that are necessary to produce the results in the next chapter. Both components are related to the input of the SK algorithm described in Section 4.5. These inputs are a covariance matrix estimate and a set of maximal cliques for a graph structure.

5.3.1 Covariance Estimation

The SK learning algorithm described in Section 4.5.1 requires a positive definite covariance matrix estimate as input [43]. The standard unbiased estimate of the covariance matrix from a data sample $\hat{\Sigma}$ defined by (4.4) is unsuitable in some cases because of the limited number of samples in our data sets. Schäfer and Strimmer point out that in the "small n, large p" case where $p \gg n$, the matrix $\hat{\Sigma}$ is not full rank, thus it is not positive definite [40]. Considering that we are learning two models, one for positive and the other for negative patients, the data set is effectively reduced in size by one half. Also, because we employ cross validation to estimate future performance, we are further reducing the data available for training each of these models. Thus, it is likely that $\hat{\Sigma}$ is not positive definite. To address this problem, PIA uses the more robust covariance estimate described by Schäfer and Strimmer [40].

This estimate is called a shrinkage estimate and has the form

$$\Sigma^\star = \lambda T + (1 - \lambda)\hat{\Sigma}$$

where T is called the target matrix and $\lambda \in [0, 1]$ is used to combine T with $\hat{\Sigma}$. Like $\hat{\Sigma}$, the target T is estimated from the data but it is chosen to have fewer parameters than $\hat{\Sigma}$ to ensure that its estimate has lower variance than $\hat{\Sigma}$ given a small number of samples. We select the target $T = I$, the $p \times p$ identity matrix, called target A by Schäfer and Strimmer [40], for its simplicity although the authors propose several others. They show that if Σ is the true covariance matrix, then the loss function $L(\lambda) = \|\Sigma^\star - \Sigma\|_2^2$ has a closed form expression for the value of $\lambda = \lambda^\star$ that minimizes it. To compute the estimate $\Sigma^\star$, we compute $\hat{\Sigma}$ from the data set and then calculate the value of $\lambda^\star$ according to this expression. The PIA system uses $\Sigma^\star$ as the input to the SK and GL algorithms described in Section 4.5. This matrix is guaranteed to be positive definite and will tend to produce an estimate that is closer to Σ than the unbiased estimate $\hat{\Sigma}$ [40].

5.3.2 Maximal Cliques

As noted above, the runtime of the SK algorithm applied to learn parameters for a GMRF model is improved considerably when given the set of maximal cliques in the underlying graph structure. PIA uses a maximal clique enumeration algorithm described by Bron and Kerbosch (BK) to find this set of cliques [8].

		nodes	edges	largest clique
all pathways	default graph	1,493	1,731	5
	fully marginalized	48	572	34
	greedily marginalized	80	259	7
	merged	55	65	4
	merged + greedily marginalized	49	59	4
cachexia pathways	default graph	386	487	4
	fully marginalized	35	284	24
	greedily marginalized	47	120	7
	merged	42	52	4
	merged + greedily marginalized	36	43	4

Table 5.4: Largest clique sizes in the human metabolic graphs after removing currency metabolites.

The BK algorithm (version one) is a simple search algorithm that maintains a fully connected set of nodes C at every search step. Specifically, every pair of nodes in C are adjacent in the graph. Initially empty, the set C may or may not be a maximal clique. Every node not in C connected to every node in C is stored in one of two sets. The first of these is the set of candidate nodes D that are the nodes that have not been considered as extensions of those in C. The other is the set of nodes R that have been considered. Initially, D contains all nodes in the graph and R is empty. Each search step takes a node v out of D and adds it to C and updates D and R to reflect the new set C. If both D and R become empty, the search branch stops and C is declared maximal or if D becomes empty but R does not then the search branch stops and C is discarded. Otherwise, the search continues until completion. At this point, the node v is placed into R and another candidate is considered.

The problem of enumerating the maximal cliques of a graph is NP-hard following a trivial reduction from the NP-complete clique problem[1] [29]. Despite the hardness of the problem, we find that the BK algorithm runs very fast for the graphs that we consider. There are other, more sophisticated algorithms that are more efficient but the BK algorithm is easy to implement. Also, the metabolic network does not change frequently, thus we can compute the maximal cliques for a graph and store them for reuse with different diagnosis tasks. Even if the maximal clique finding were a bottleneck, it is rarely necessary to perform the computation. In our experience, the improvement in speed of the SK algorithm is worth this computational effort.

After applying the maximal clique finding algorithm to the graph structures listed in Table 5.3, a distinct feature of the fully marginalized graphs becomes apparent. The size of the largest clique in these graphs is very large, containing most of the nodes in the graph. The largest clique sizes for the human graphs are given in Table 5.4. Unfortunately, these clique sizes imply that much of the pathway structure from the default graphs is lost in the transformation. The fully marginalized graphs are the only graphs that contain only observed variable nodes. We consider the influence of this feature on the results in the next chapter.

[1] Given an undirected graph G and a positive integer k, determine whether G contains a clique of size k.

Chapter 6

Results

This chapter describes the results of applying the ideas discussed in previous chapters to a variety of classification tasks derived from three data sets. We describe various experiments that we performed with PIA and summarize the results we obtained with the system on these tasks. More detailed numerical results from these experiments are provided in Appendix B.

6.1 Data Sets

We use several data sets to evaluate the learning algorithms described in this thesis. This section describes the relevant properties of these data sets including the various classification tasks that we derive from them. Our main data set contains metabolic profiles of human cancer patients with and without cachexia. In addition, the data set contains healthy human metabolic profiles. We also have a data set of worm metabolic profiles with many associated classification tasks. Worms in this data set were subjected to various dietary and genetic manipulations that we use classifiers to predict from metabolic profile measurements. Another data set was produced by a study to determine the amino acid intake requirements of pigs. Based on these metabolic profiles, we learn classifiers that predict whether the intake of a pig is above or below this requirement. For reasons discussed below, the data from this study is produced by a biological process similar to cachexia.

6.1.1 Cachexia Data Set

The cachexia data set contains metabolic profiles for 148 people, 38 of whom are healthy and 110 have cancer. Of the 110 cancer patients, we ignore four because of kidney failure leaving a total of 106. Each metabolic profile quantifies the concentrations of 64 metabolites. The metabolite urea is excluded from the data set because its quantification is unreliable [39]. Each person in the data set is described by the concentrations of 63 metabolites in a spot urine sample. The data set contains no missing values; every metabolite was quantified for every patient. The raw metabolite concentration values were all log transformed (using base e) before any analysis was performed on them. The cachexia data set also contains spectral binning data for each cancer patient with 206 bins and no missing values.

A pair of abdominal CT scans were performed on 93 of the 106 cancer patients approximately three months apart. These scans were analyzed to determine the percentage of muscle mass that the patient lost (or gained) per 100 days. Thus, there is a real number associated with each patient quantifying their change in muscle mass. Ideally, we would produce a regression model that could predict this particular value. Instead, we focus on the easier classification task of determining whether or not a patient is cachexic. We define the disease state of a patient based on their muscle loss percentage.

	instances	positive	negative	base accuracy
cachexia diagnosis	73	44	29	60.3
cancer diagnosis	144	106	38	73.6
cancer type	106	35	71	67.0
sex	106	41	65	61.3

Table 6.1: Classification tasks derived from the cachexia data set and their properties.

Essentially, if a patient lost muscle we define them as cachexic and if they gained muscle then they are not. The instrument that detects the muscle mass changes is only precise to $\pm 0.75\%$ per 100 days however. Therefore we define a patient as cachexic if their muscle loss value is $\geq 0.75\%$ and as not cachexic if the value is $\leq -0.75\%$. Patients with loss values in the interval $(-0.75, 0.75)$ are ignored in the classification task. The result of this definition of the two classes is a data set with 73 patients. Of them, 44 have cachexia and 29 do not. Because 60.3% of the patients are positive, we expect the accuracy of the predictions made by our classifiers to exceed this rate. A simple classifier that always predicts positive trivially achieves 60.3% accuracy.

In addition to the muscle loss values that are used to define the disease classes, the data set contains other information from which to derive additional classification tasks that we use to test our methods. The 106 cancer patients in the data set have one of two cancer types: colorectal cancer (35 patients) or lung cancer (71 patients). Thus, we consider predicting the type of cancer that a patient has as another task. Also, 38 people in the data set are healthy so we will consider the general cancer diagnosis task. The sex of the cancer patients is also recorded in the data set so we consider predicting this attribute as well. Details of the four classification tasks are summarized in Table 6.1. Naturally, each task has a baseline accuracy rate that a simple classifier that always predicts the majority class can achieve. The learned classifiers should perform at least as well as this.

6.1.2 Worm Data Set

This data set contains 57 metabolic profiles from worms (C. elegans) each quantifying 30 metabolites. These metabolic profiles are derived from samples composed of the entire body of the worm. Thus, the profiles more closely reflect the state of the organism than the urinary metabolic profiles of the cachexia data set. This data set has a different problem that is the consequence of many zero value measurements in it. For certain classification tasks derived from it, there are metabolites with no nonzero values in the subset of the data specific to a particular class. Thus, we cannot produce the required positive definite covariance matrix estimate for the distribution corresponding to that class. To avoid these problems, metabolites with ten or more zero values in the data set are dropped. There are four such metabolites, leaving a data set with 26 measurements per worm. Because we are only concerned with comparing classifiers to each other and not achieving the best possible classifier performance, we consider this simple approach to making the data set amenable to our analysis acceptable. Also, the data set was already log transformed as provided, thus we did not perform this preprocessing step.

Worms in the data set are from one of three strains: the wild type or one of two mutant strains. The mutant strains are produced by a gene knockout procedure that deactivates a specific enzyme gene. We denote the set of wild type worms by WT and the sets of mutant worms by $KO1$ and $KO2$. Further, each of the worms is fed one of two diets producing the two diet-specific worm sets $D1$ and $D2$. Finally, of the worms in $D2$, some are subjected to one of two gene knockdown procedures that reduce the expression of a particular gene. The sets of worms subjected to knockdown 1 and knockdown 2 are denoted by $KD1$ and $KD2$

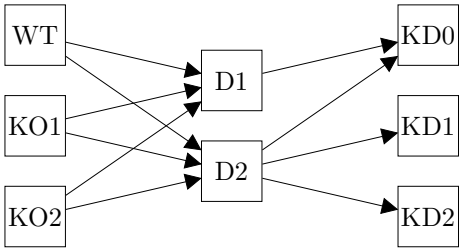

Figure 6.1: Possible characterizations of the worms in the worm metabolic profile data set.

	instances	positive	negative	base accuracy
wild type vs. others	57	22	35	61.4
knockout 1 vs. others	57	17	40	70.2
knockout 2 vs. others	57	18	39	68.4
knockdown vs. no knockdown	57	25	32	56.1
diet 1 vs. diet 2	57	15	42	73.7

Table 6.2: Classification tasks derived from the worm data set and their properties.

respectively and the remaining worms, subjected to no knockdown, are denoted $KD0$. There are many possible categorizations for the worms in this data set. These are summarized by the possible paths through the graph in Figure 6.1. Every worm must have a strain and a diet but they are not necessarily subjected to a gene knockdown.

We derive several binary classification tasks from the possible worm categorizations. Although tasks with more than two classes are possible and possibly make more sense from a biological perspective, our software was initially developed to support binary classification tasks only. As noted, we are not concerned with the optimal classification performance on this particular data set. Because we are only interested in evaluating the performance of PIA relative to standard machine learning algorithms, we did not modify the software to support more general classification problems. The tasks that we consider and their relevant properties are summarized in Table 6.2.

6.1.3 Pig Data Set

The pig (S. scrofa) data consists of four related data sets. These data sets were produced by studies to determine the required intake of two different amino acids for pigs [42, 41]. The two amino acids cysteine (Cys) and methionine (Met) were fed to the pigs in varying amounts to quantify these requirements. Pig diets were delivered via two different routes: intravenous (IV) and intragastric (IG). The four data sets were produced from the four possible combinations of amino acid and route.

Each of the two amino acids has a required intake quantity that depends on the route that the compound is fed. Pigs in the four data sets have a value corresponding to the amount of Cys or Met in their diet. These amount are either above or below the required amount, thus we have a natural classification task to consider for each data set. Specifically, the classifiers predict whether or not the diet of a pig contains the required amount of an amino acid from its metabolic profile. Relevant details of these classification tasks are summarized in Table 6.3. Metabolic profiles in the Cys and Met data sets quantify 26 and 25 amino acid metabolites respectively from a blood sample taken from each pig. As with

	instances	positive	negative	base accuracy
Cys, IG	27	12	15	55.6
Cys, IV	28	13	15	53.6
Met, IG	28	16	12	57.1
Met, IV	32	17	15	53.1

Table 6.3: Classification tasks derived from the pig data set and their properties.

the human metabolic profile data set, we log transform the pig data (using base e) before analysis.

In a sense, the tasks derived from the pig data sets are similar to cachexia diagnosis. One of the main metabolic characteristics of cachexia is the catabolism of the amino acids produced by the breakdown of muscle tissue. Patients with cachexia tend to have more amino acids available than those without cachexia. These amino acids are either used to build proteins or they are catabolized [37]. They are not stored for future use. Because the construction of a protein depends on many different amino acids, if one of these amino acids is not available in sufficient quantity, protein production is limited. The other amino acids that would have been used to make proteins are no longer usable, thus they are catabolized. This process is induced by holding the intake of a particular amino acid in the diets of pigs below the required level. Therefore, the byproducts of amino acid catabolism should be reflected in the metabolic profiles of these pigs in a manner similar to that of human cachexia patients.

6.2 Classification Results

This section presents classification results for the tasks and the algorithms described above. KEGG contains metabolic pathways for the three organisms that are used to derive our classification tasks, thus the PIA algorithms all use organism-specific KEGG pathways. In addition to using the metabolic pathways to derive fixed structures for PIA models, we can also incorporate them into GL. This involves using a matrix penalty in the GL optimization based on a known graph as described in Section 4.5.2. In this section we abbreviate these results as PIA+x or GL+x where x is one of

- FM = fully marginalized structure
- GM = greedily marginalized structure
- MM = merged, then greedily marginalized structure

indicating the transformation performed on the underlying graph. These transformations are described in detail in Section 5.2.

There are two other relevant dimensions to the results presented in this section. First, we compare the full set of KEGG pathways with a subset of them. This subset, designated as the cachexia pathways in Section 5.1, are those pathways deemed to be relevant to the metabolic processes underlying cachexia. The list of these pathways is given in Table 5.1. Second, the set of pathways also affects what metabolites are actually included in the PIA models. In a given data set, not every metabolite actually appears on a KEGG metabolic pathway. The measured metabolites for the three data organisms and the corresponding number of those metabolites on a pathway are given in Table 6.4. We treat these KEGG-selected metabolites as a form of feature selection and evaluate classifiers on data sets restricted to them.

To obtain the results in this section we use five fold CV for all algorithms and all tasks. The instances in each fold are selected to match the class distribution in the full

	total metabolites	KEGG-selected metabolites
human	63	48
worm	26	19
pig	26/25	14

Table 6.4: Total number of metabolites measured for each organism and the number of them appearing on an organism-specific KEGG metabolic pathway.

data set as closely as possible. For a given task, the same division of instances into folds is used for every algorithm. The performance metric used to evaluate these algorithms is classification accuracy. Classifier performance is measured by averaging its accuracy over the five folds (even though they may not have equal size). To determine if the difference in performance between two classifiers on the same task is significant we apply a paired t-test. Each algorithm has five accuracy values associated with each of the five folds in a task. We treat these values as independent random variables that depend on a specific classifier. The paired t-test determines if the difference in the mean of these values for two classifiers is significant. We consider a difference to be significant if the p value from the test satisfies $p < 0.05$.

The implementations of the algorithms presented here are from a variety of sources. SVM results are based on LibSVM with a radial basis function kernel using the default parameters [11]. The C4.5 implementation is provided by Quinlan [38] and our results with it are based on the default settings as well. Friedman et al. provide an implementation of their GL algorithm [19]. To select the value of the regularization term ρ we use an internal five fold CV technique over the values $\rho \in \{0.01, 0.1, 0.2, \ldots, 1.0\}$. After determining the best choice of ρ from this set on the training data, we apply it to the full training set to learn a model. The remainder of the algorithms used here were implemented by the author based on their descriptions in the literature. Chapter 4 discusses these algorithms in detail and provides references for them.

6.2.1 Cachexia Data Set Results

Results for the cachexia diagnosis task are given in Figure 6.2. Error bars indicate one standard deviation of the fold accuracy values in this and other figures presenting classification accuracy results. The base classification accuracy for each task is indicated by the horizontal line in these figures. The cachexia diagnosis results show that the PIA model based on all human metabolic pathways with the full marginalization transformation applied produces the best classification accuracy. Although promising, the difference between this result and some of the standard algorithms is not statistically significant. Results for the other classification tasks are given in Figures 6.3, 6.4 and 6.5. In the these tasks, none of the PIA models outperforms the standard algorithms although the performance of PIA is comparable to them in all cases. The exact classification accuracy of each algorithm for the four tasks in the cachexia data set are given in Table B.1. Accuracy results for spectral binning data is given in Table B.12.

Along the other dimensions that we considered we see no significant differences as well. In all cases, PIA models based on all metabolic pathways slightly outperform the corresponding model based on the cachexia pathways. The improvement in classification accuracy of PIA at cachexia diagnosis obtained by restricting the system to the relevant cachexia pathways is shown in Figure 6.6. Detailed PIA results with cachexia pathways for the all four tasks are given in Table B.10. PIA models based on the cachexia pathways are only strictly better for the sex classification task. When the standard machine learning algorithms are applied to data sets based on all metabolites compared to KEGG-selected metabolites, no

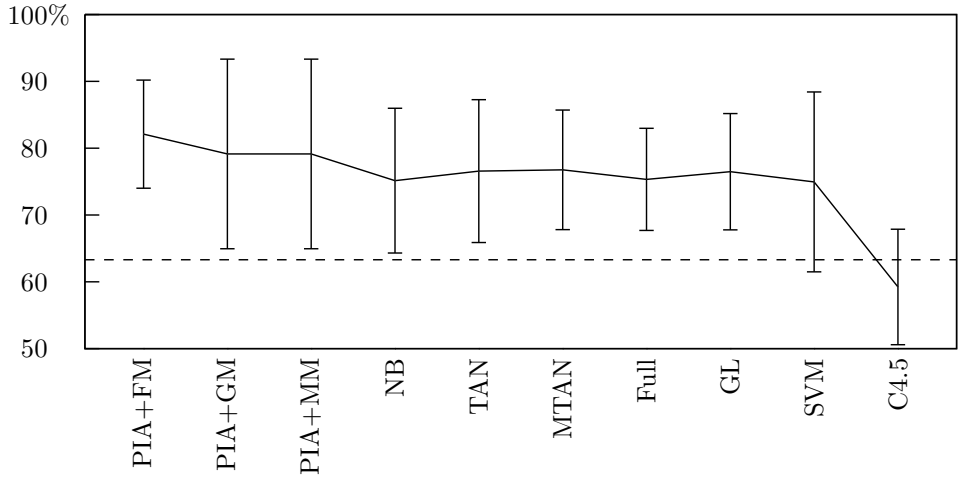

Figure 6.2: Classification accuracy results for the cachexia data set, cachexia diagnosis task.

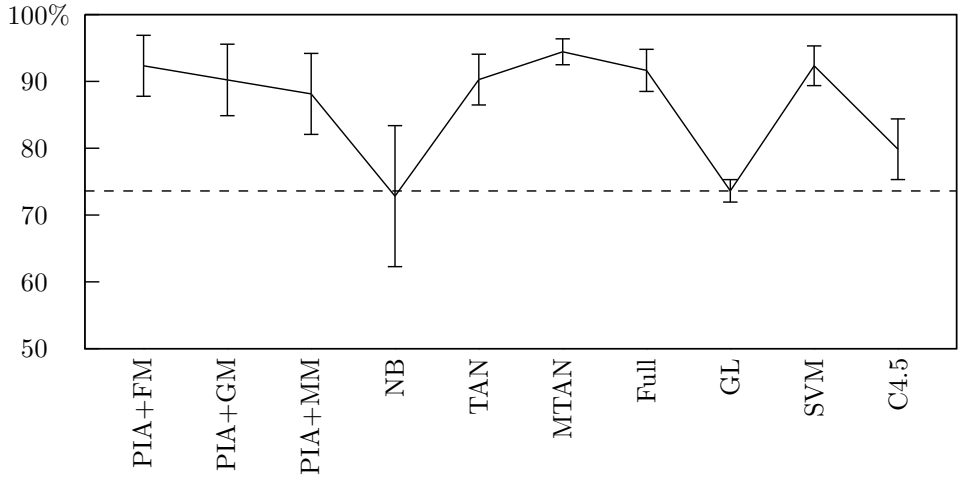

Figure 6.3: Classification accuracy results for the cachexia data set, cancer diagnosis task.

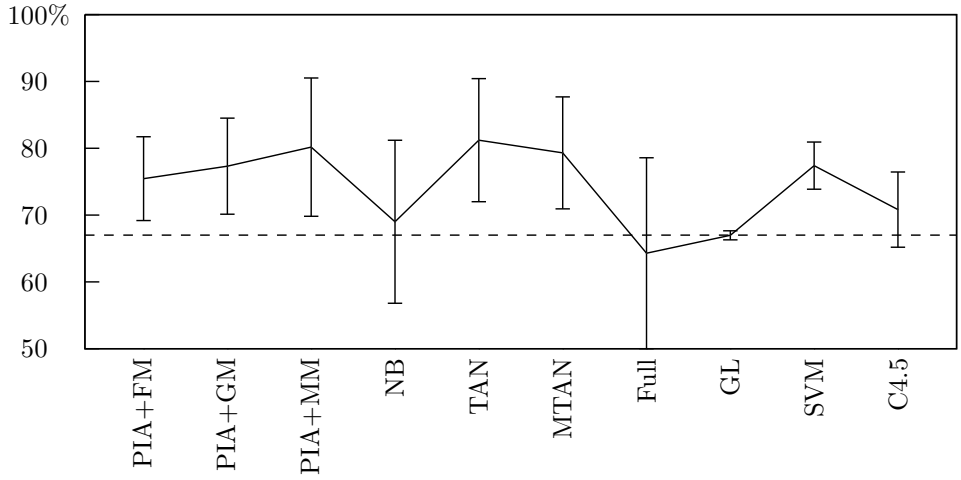

Figure 6.4: Classification accuracy results for the cachexia data set, cancer type task.

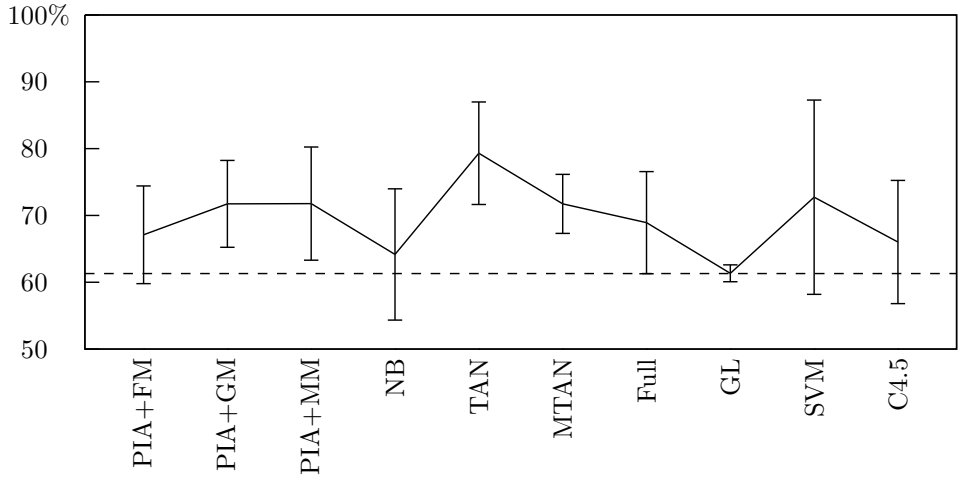

Figure 6.5: Classification accuracy results for the cachexia data set, sex task.

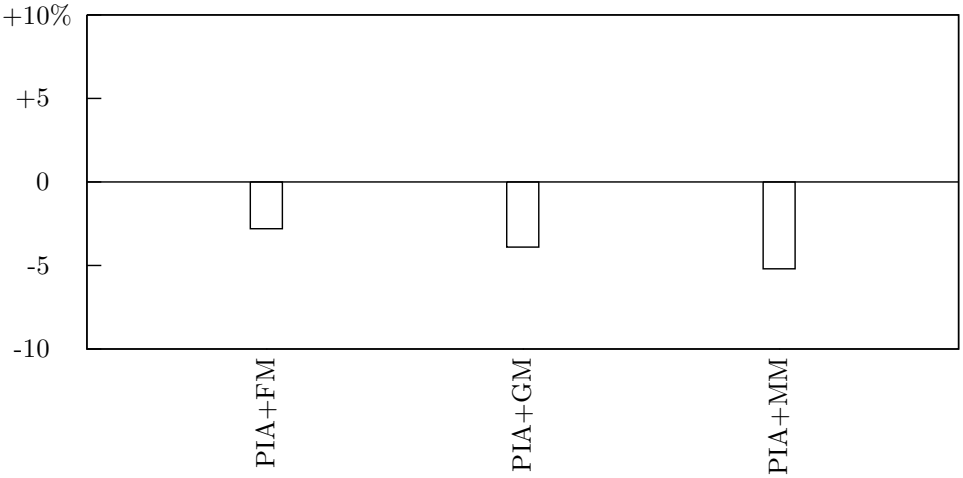

Figure 6.6: Classification accuracy improvements obtained by models limited to cachexia-relevant pathways for the cachexia data set, cachexia diagnosis task.

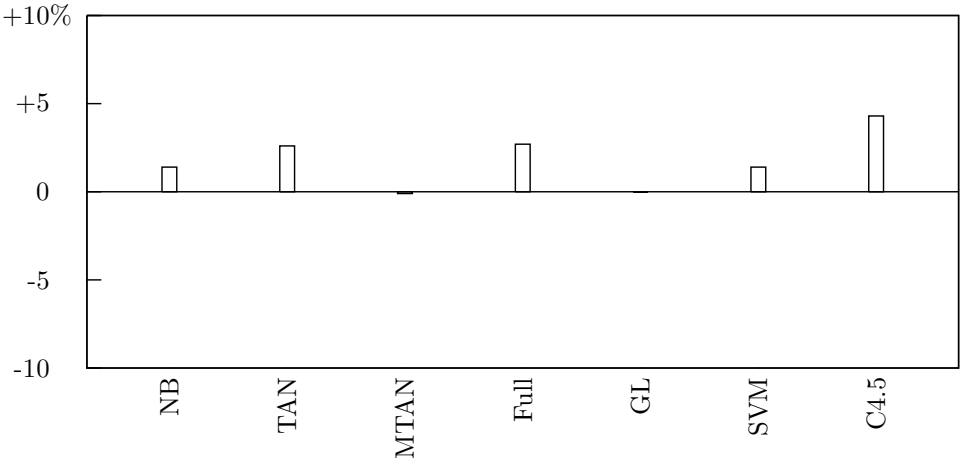

Figure 6.7: Classification accuracy improvements obtained by models limited to KEGG-selected metabolites for the cachexia data set, cachexia diagnosis task.

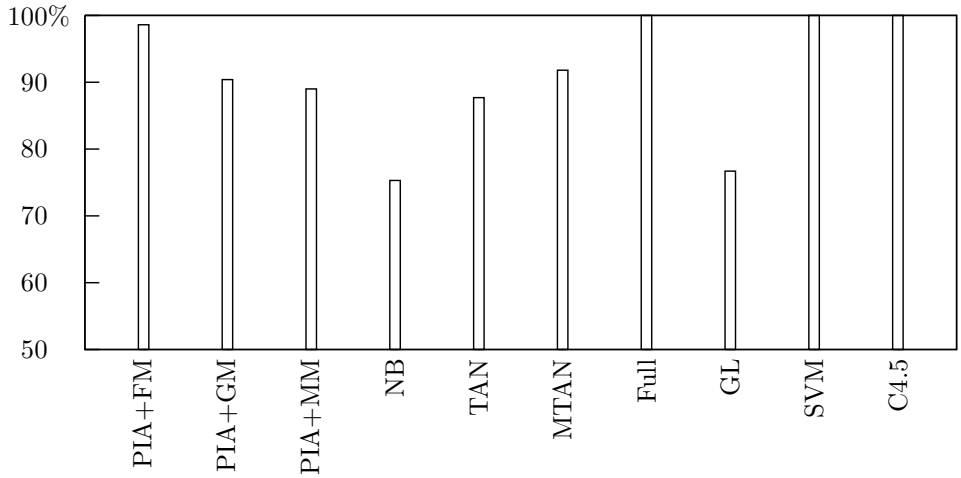

Figure 6.8: Classification accuracy results on the training data for the cachexia data set, cachexia diagnosis task.

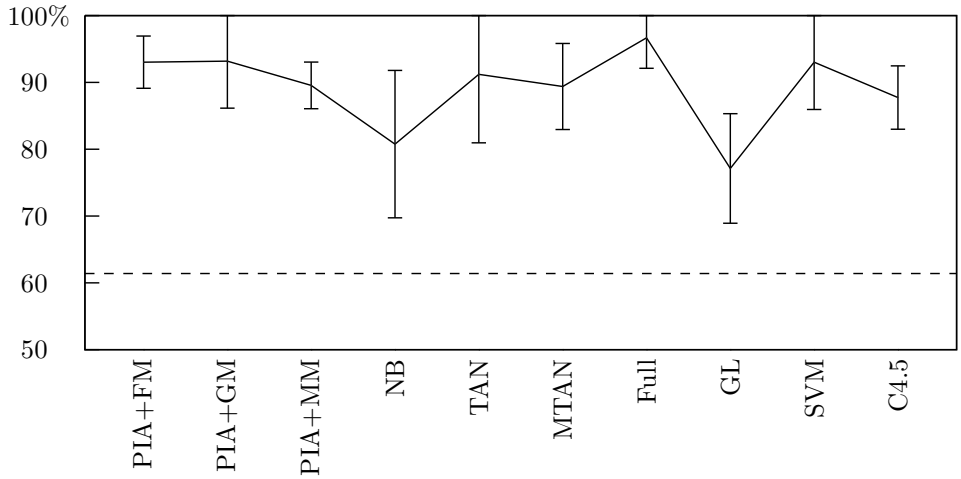

Figure 6.9: Classification accuracy results for the worm data set, wild type task.

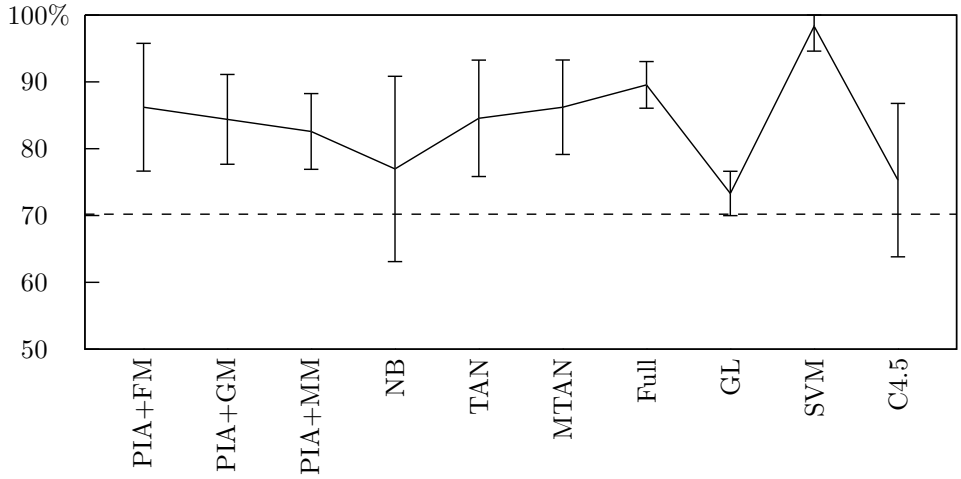

Figure 6.10: Classification accuracy results for the worm data set, knockout 1 task.

significant differences are observed either. Classification accuracy improvements for the cachexia diagnosis task with KEGG-selected metabolites only are shown in Figure 6.7. Detailed results for all four tasks are given in Table B.7.

These results provide no evidence that any form of expert-provided knowledge improves classification accuracy on human metabolic profile data. Standard approaches that are better developed and more widely used are probably as good as or better than the PIA model. One possible explanation for these results is that the PIA models overfit the data. To measure this, we use each algorithm to learn a classifier using the entire data set as a training set and then evaluate it on the same data. Models with high classification accuracy in this setting are more likely to have overfit the data than those with lower accuracy rates. The training set results produced by all algorithms on the cachexia diagnosis task are shown in Figure 6.8. Results from all tasks are given in Table B.4. As indicated, PIA models are among the most overfit of the algorithms that we considered.

6.2.2 Worm and Pig Data Set Results

The conclusions reached based on the cachexia data set are not improved when we consider the other two data sets. Classification accuracy results for the tasks derived from the worm data set are shown in Figures 6.9, 6.10, 6.11, 6.12 and 6.13. Detailed numerical results are given in Table B.2. Accuracy results for the pig data set classification tasks are in Figures 6.14, 6.15, 6.16 and 6.17. Table B.3 contains the detailed results. As with the cachexia data set results, the PIA models are comparable to the standard algorithms on these tasks but they are not significantly better.

KEGG-based feature selection does not appear to make a difference for the classification accuracy of the standard algorithms either. The improvements obtained in this setting for the worm and pig data set classification tasks are given in Tables B.8 and B.9 respectively. In most cases, classification accuracy is reduced when classifiers are restricted to metabolites that appear on a KEGG metabolic pathway. For the Met IG and Met IV tasks from the pig data set, KEGG-based feature selection produces improvements in classification accuracy that are approximately 15% better for classifiers based on graphical models (GL, NB, TAN, MTAN and Full).

As with the cachexia data set, we measured the training set error for each classifier on the worm and pig classification tasks. These results are given in Tables B.5 and B.6 for these two sets of tasks respectively. As with the cachexia data set, PIA models tend to exhibit a high degree of overfitting relative to some of the standard approaches.

6.3 Other Results

We consider some other situations to determine if there is any benefit to using KEGG metabolic pathways in a metabolic profile classifier. This section goes beyond the simple classification experiments presented in Section 6.2 to investigate the performance of PIA in these situations. The focus of this section is on the cachexia data set only.

6.3.1 Additional Variables

As previously discussed, only a subset of the metabolites in any of our data sets actually appears on a KEGG pathway. This may limit the accuracy that PIA can achieve relative to the other classifiers because it is based on less data than them. In this section, we consider ways to overcome this limitation by incorporating additional variables into the PIA models.

There are three approaches to incorporating a variable into a pathway-based GMRF model when that variable does not correspond to a metabolite on one of those pathways. The first assumes that the variable is completely independent of every other variable. Thus,

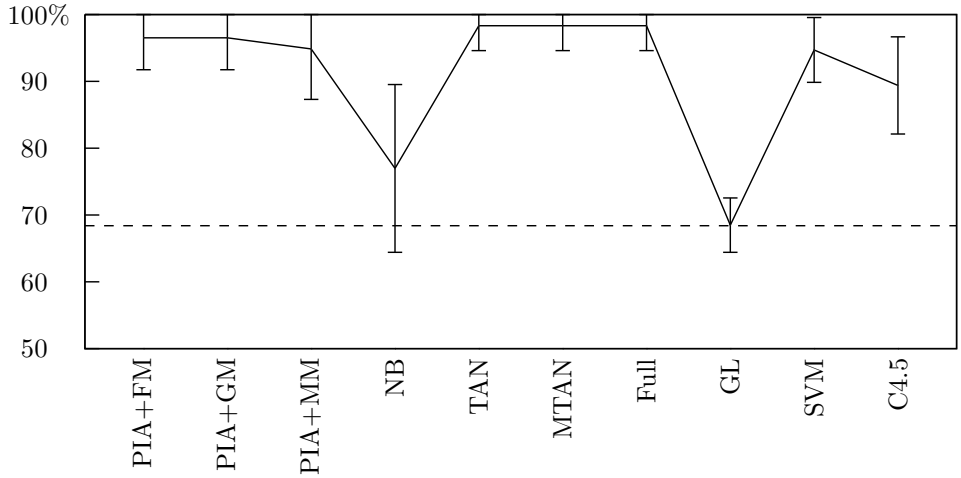

Figure 6.11: Classification accuracy results for the worm data set, knockout 2 task.

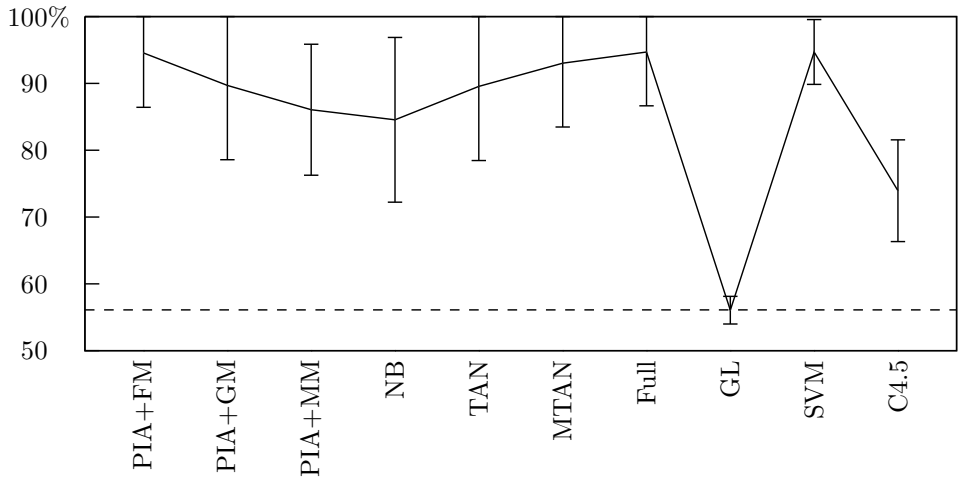

Figure 6.12: Classification accuracy results for the worm data set, knockdown task.

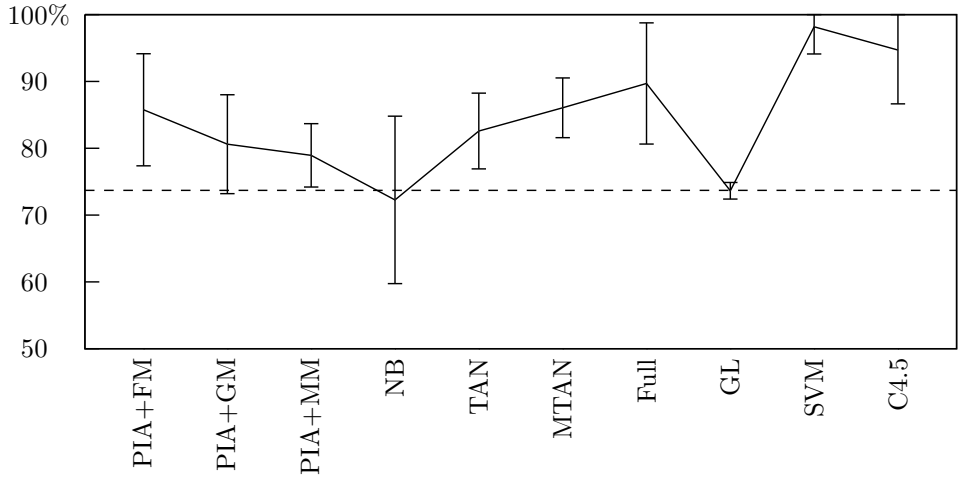

Figure 6.13: Classification accuracy results for the worm data set, diet task.

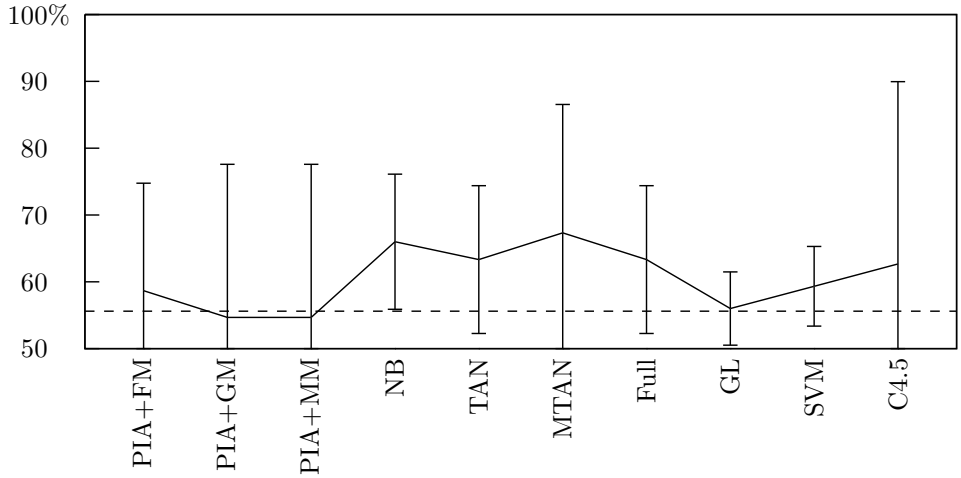

Figure 6.14: Classification accuracy results for the pig data set, Cys IG task.

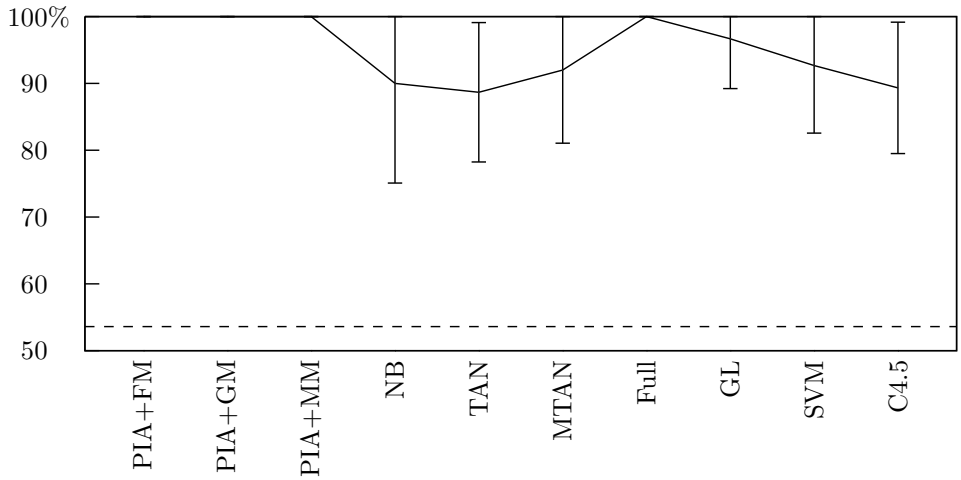

Figure 6.15: Classification accuracy results for the pig data set, Cys IV task.

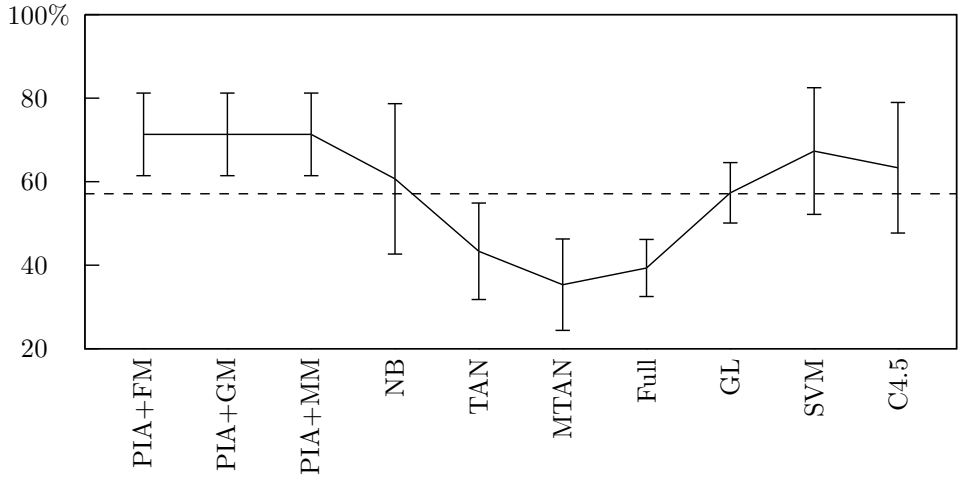

Figure 6.16: Classification accuracy results for the pig data set, Met IG task.

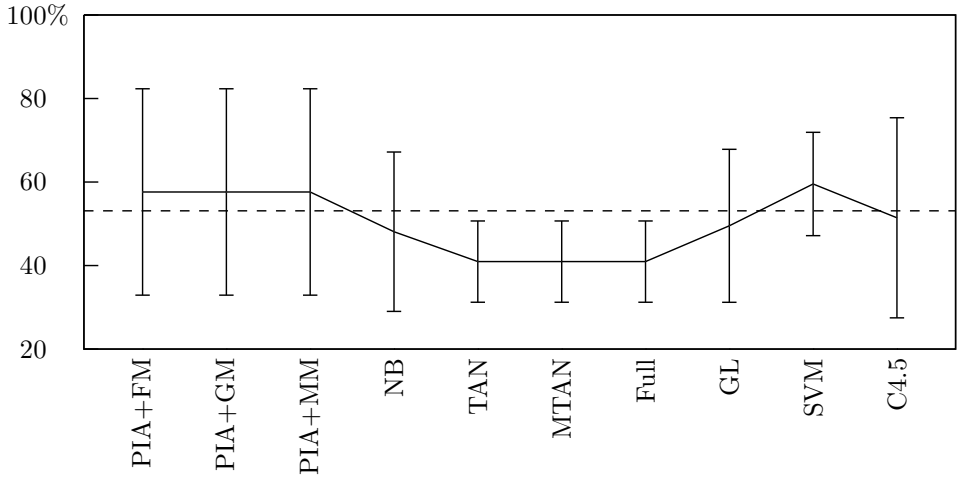

Figure 6.17: Classification accuracy results for the pig data set, Met IV task.

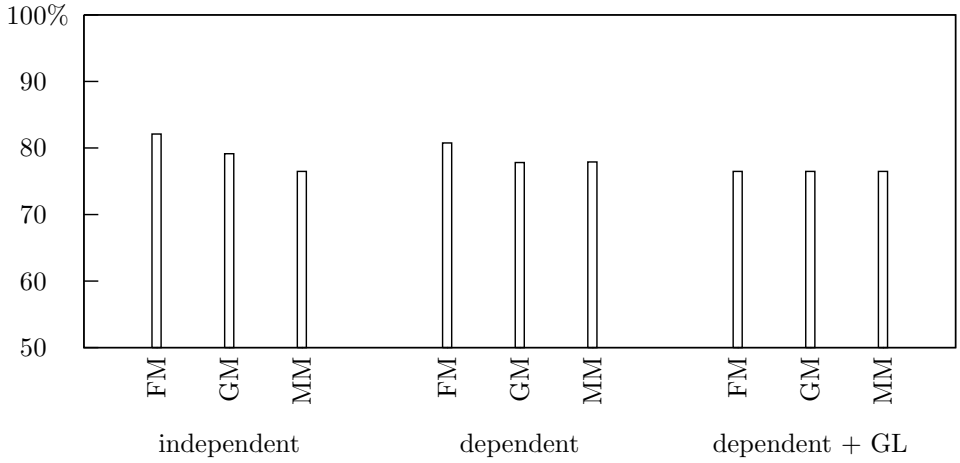

Figure 6.18: Classification accuracy results obtained by models extended with age and sex variables for the cachexia data set, cachexia diagnosis task.

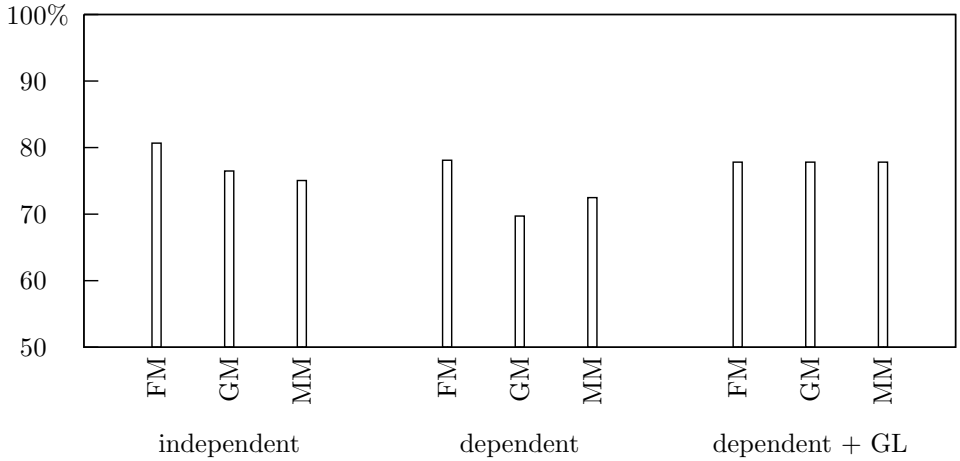

Figure 6.19: Classification accuracy results obtained by models extended with additional metabolite variables for the cachexia data set, cachexia diagnosis task.

the node corresponding to the added variable will have no incident edges in the graph structure. Another approach assumes the opposite: that the variable is directly dependent on all of the others. In this case, the added node is adjacent to all other nodes in the graph structure. A more complicated approach is to fully connect nodes corresponding to added variables as in the latter case. We can then use the GL algorithm to learn the correct way to connect the new nodes to the others. Note that GL may also remove connections among existing nodes.

In addition to the extra metabolites not found on a KEGG pathway, the cachexia data set contains other information about the patients in it. This information includes age and sex, two factors that we know influence metabolite concentrations. In general, we would like a way to incorporate such additional information into the PIA system because it may be relevant to a diagnosis task of interest. Perhaps the easiest approach is to assume the additional variables follow normal distributions and to add them into the model using one of the approaches just suggested. In this case, we encode sex as a variable with numeric values zero and one. This method is not correct because, particularly with age and sex, the additional information is not likely to follow a normal distribution. The method is easy to implement however. Additional methods are an important area of future work with PIA because some of the variables we might consider adding to the models could be very important for distinguishing disease states.

We consider six experiments based on these ideas:

- additional metabolites, independent
- additional metabolites, dependent
- additional metabolites, dependent + GL
- age and sex, independent
- age and sex, dependent
- age and sex, dependent + GL

The classification accuracy results for PIA models extended with age and sex applied to the cachexia diagnosis task are shown in Figure 6.18. The same results for models extended with additional metabolite variables are shown in Figure 6.19. Numerical results all of these models are given in Table B.11. These results make it clear that we cannot improve on the best results obtained without additional variables. Models based on GL all obtain the same performance as they did without additional variables. The extended models for the MM and GM transformations improve slightly over the original models when the additional variables are added independent of the others. The difference here is not statistically significant however. In general, the independent models outperform the dependent models suggesting that this approach is the better of the two.

6.3.2 Patient Subsets

Another dimension that we investigated to determine if PIA has any advantage is learning from subsets of the patients in the data set. Our hypothesis is that PIA might have an advantage when the number of examples available for learning is small. In this case, knowing about the pathways might help produce better models of the data that more accurately classify future patients. Since the results given above indicate similar accuracy rates for PIA models and the standard machine learning algorithms, it is possible that the cachexia data set already contains enough patients for any reasonable algorithm to obtain the best possible accuracy. There are two approaches that we consider to investigate the hypothesis.

One way to investigate the hypothesis that PIA has an advantage when data is limited is to consider random subsets of the data set of various sizes. Here we still use a muscle loss threshold of ± 0.75 to define the classes. We did this for subset sizes in the range $12, 11, \ldots, n$ always following the class distribution of the full data set in the subsets that we consider as closely as possible. Starting the subset sizes at 12 ensures that every data set considered contains at least five patients from each class. For each size, we perform LOOCV with 100 random subsets of that size and average the results and compute the standard deviation. Results comparing PIA+FM with NB, TAN, MTAN and Full are given in Figures 6.20, 6.21, 6.22 and 6.23 respectively. The PIA+FM results almost always fall within one standard deviation of the considered algorithms regardless of the number of patients used in the small data sets. Only when most patients are included does a significant difference emerge because the standard deviation approaches zero. As indicated above, the differences at the upper end of the size range are not significant.

We consider another approach to generating smaller data sets based on changing our definition of cachexia. As noted above, the patients in the cachexia data set each have a real number quantifying their loss in muscle mass. We used this variable l_r to divide the patients into two classes: $c_r = +$ if $l_r \geq 0.75$, $c_r = -$ if $l_r \leq -0.75$ and the patient is dropped otherwise. The threshold $t = \pm 0.75$ is not the only one we could consider. Every such threshold drops more or fewer patients from the effective data set. To validate the hypothesis in this case, we considered varying the threshold t in the following manner. The threshold t is started at zero and gradually increased in magnitude, dropping more and more patients from the data set until not enough remain to compute a positive definite covariance estimate. Again, we use LOOCV to evaluate the algorithms with the generalized definition of the classes. As with the previous set of results, we observe no significant difference between PIA+FM and the four standard algorithms NB, TAN, MTAN and Full.

6.3.3 Edge Priors

Up to this point, in all of the experiments that we have considered, the metabolic pathway graph structures derived from KEGG were treated as fixed structures. This means that if two metabolites are not joined via an edge, then PIA will never join them when learning regardless of the information contained in the training set. Recall that the GL learning algorithm may remove an edge from an initial structure with some probability that depends on the value of the parameter ρ. As described in Section 4.5.2, we can generalize ρ so that each possible edge $\{i, j\}$ has a value ρ_{ij} specific to it. In this way we can influence the probability that a particular edge is included in a learned structure. This allows us to treat a metabolic pathway graph structure as a soft structure: edges implied by KEGG are given a higher probability of inclusion but no edge is ruled out completely.

We achieve this by considering two parameter values instead of one. The first, denoted ρ_e is the value assigned to ρ_{ij} for every edge $\{i, j\} \in E$ in a graph $G = (V, E)$ derived from KEGG. The other parameter is $\rho_{\overline{e}}$ is assigned to ρ_{ij} for every $\{i, j\} \notin E$. Values of ρ_e and $\rho_{\overline{e}}$ are selected from the set $\{0.01, 0.1, 0.2, \ldots, 1.0\}$ such that $\rho_{\overline{e}} > \rho_e$. We consider all such combinations and perform five fold CV with GL learning and starting with a KEGG structure. Classification accuracy results for the cachexia diagnosis task staring with graphs produced via the FM, GM and MM transformations are given in Tables B.13, B.14 and B.15. These results show that we obtain a small but insignificant improvement with this technique when the FM or GM transformations are employed. We note that selecting the best values for ρ_e and $\rho_{\overline{e}}$ by examining the results in these tables is inappropriate. The correct way to do this we should use cross validation to select the parameter values for a specific training set. Because we did not obtain a significant improvement in classification accuracy we did not pursue this approach.

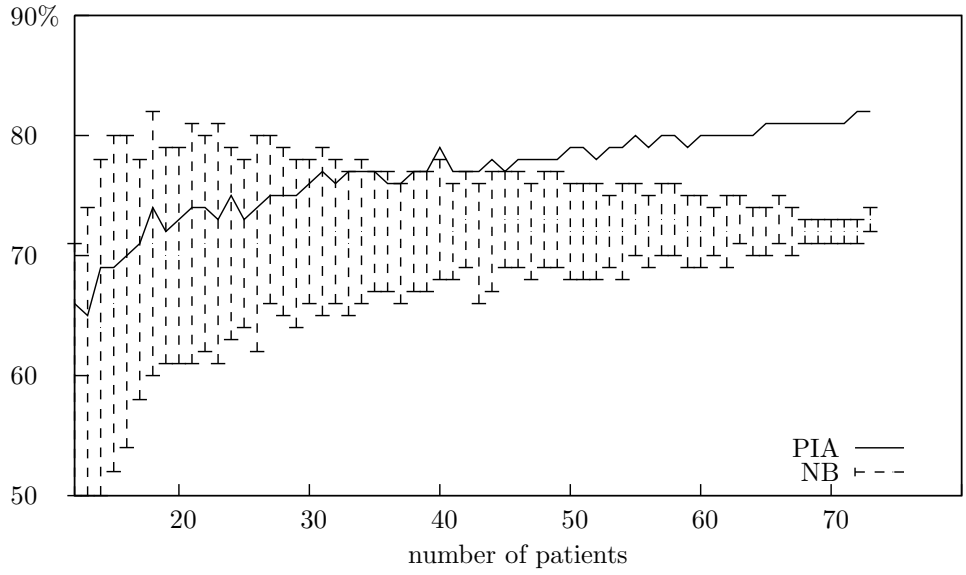

Figure 6.20: Average classification accuracy results comparing NB and PIA on random patient subsets of various sizes for the cachexia data set, cachexia diagnosis task.

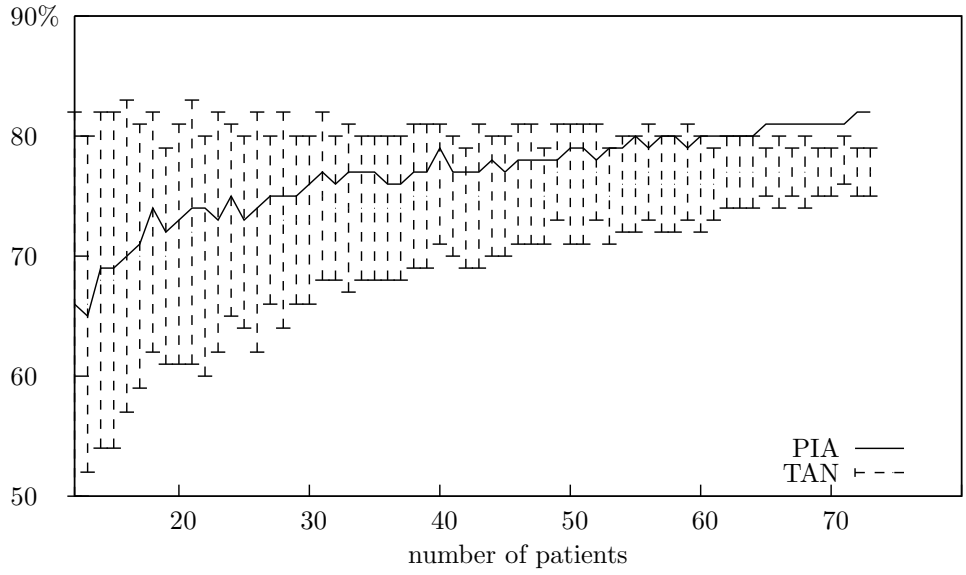

Figure 6.21: Average classification accuracy results comparing TAN and PIA on random patient subsets of various sizes for the cachexia data set, cachexia diagnosis task.

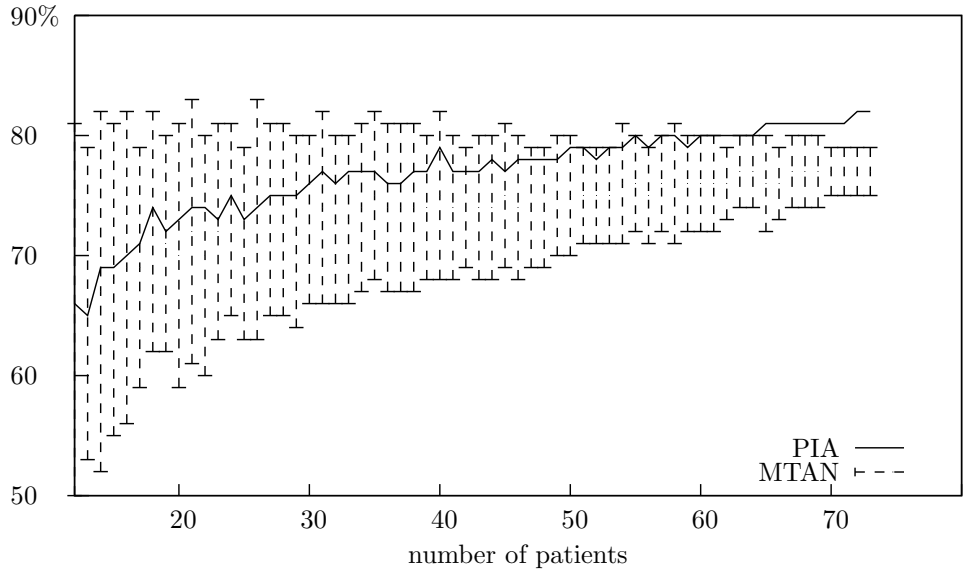

Figure 6.22: Average classification accuracy results comparing MTAN and PIA on random patient subsets of various sizes for the cachexia data set, cachexia diagnosis task.

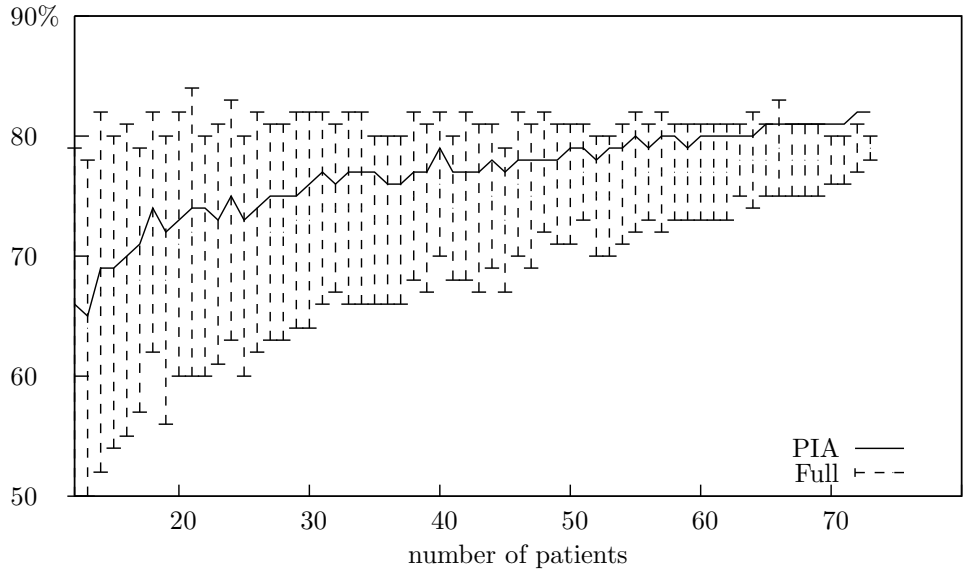

Figure 6.23: Average classification accuracy results comparing Full and PIA on random patient subsets of various sizes for the cachexia data set, cachexia diagnosis task.

6.4 Conclusion

The results presented in this chapter suggest that incorporating metabolic pathway knowledge into classifiers based on metabolic profile data provides no benefit in terms of the accuracy that those classifiers can achieve. The fact that the performance of the PIA models is comparable with standard machine learning algorithms suggests that metabolic pathway knowledge does not hurt performance either. In many cases, the differences between the performance of the algorithms is not statistically significant. This suggests that most algorithms are able to achieve the best possible classification accuracy for the tasks we consider. If true, then the prior biological knowledge could not have improved performance. Because we consider a wide variety of tasks and situations in evaluating PIA, this is unlikely however. In the case that we have not achieved the best possible classification accuracy in the tasks considered here, there are two potentially significant limitations on the performance of PIA.

The first limitation is that much of the structure of the metabolic network is lost after transforming it to reduce the number of latent variables in the resulting models. This is particularly true with the fully marginalized graph structures as suggested by Table 5.4. In these graphs, the maximal clique contains most of the nodes. Structures produced from the other transformations have the distinguishing characteristic of a single latent variable node with many measured variable nodes connected to it. In this case, marginalizing the latent variable would result in a structure with a single large clique like the fully marginalized structure. Increasing the number of measured metabolites in a metabolic profile would reduce the amount of lost structural information because the transformations are necessarily applied to remove unmeasured metabolites. Less destructive transformations may be possible as well.

Another limiting factor is the assumption that class-conditional metabolic profiles follow multivariate normal distributions. Under this assumption, PIA benefits from computationally efficient learning and classification of unseen patients. Standard machine learning algorithms such as SVM and C4.5 do not make assumptions about the distribution of feature values. It is not likely that metabolic profile measurements actually follow multivariate normal distributions. Thus, PIA may benefit from assuming a different distribution although we are not aware of any with the desirable properties of multivariate normal distributions. These include computational efficiency and ability to incorporate prior knowledge about the distribution. Although other assumptions were mentioned in Section 1.3, these are related to the way metabolic pathway knowledge is interpreted in the context of graphical models. We believe that any classifier that attempts to leverage pathway knowledge would have to make these assumptions. Thus, they are less likely to account for the inability of PIA to outperform standard machine learning algorithms.

Appendix A

KEGG Details

This appendix describes the parts of the KEGG database that PIA uses. The discussion includes how the relevant data is retrieved, its organization and its structure. We also discuss some changes that we made to the KEGG database to correct errors and to include additional knowledge of human metabolism. The KEGG data files that we use correspond to the version of the database that was available via the KEGG File Transfer Protocol (FTP) server[1] on 22 June 2009. Because changes are made to KEGG on a daily basis, the information here may not be valid for other versions of the database. As noted in Section 2.3, KEGG contains information covering many aspects of biology. The discussion here is limited to the components of the KEGG database necessary to generate a GMRF graph structure for PIA from an organism, a data set of metabolic profiles and and a set of metabolic pathways.

A.1 Metabolic Pathway Organization

KEGG data is stored in numerous flat files provided for download via FTP. The files that are relevant to the organization of the metabolic pathways used in this thesis are

```
ligand/reaction/reaction_mapformula.lst
pathway/organisms/hsa/hsa*.rn
pathway/organisms/cel/cel*.rn
pathway/organisms/ssc/ssc*.rn
```

where the `*` character corresponds to a possible KEGG pathway ID number. File names are relative to the root of the KEGG directory structure at `ftp://ftp.genome.jp/pub/kegg/`. The file `ligand/reaction/reaction_mapformula.lst` is the most important of these for PIA because it describes how chemical reaction equations map onto KEGG pathway images. KEGG pathways are commonly presented as graphical images such as the Citric Acid Cycle pathway in Figure 2.2. The reaction mapping file is an easily parsed representation of these images. This file is necessary because PIA cannot easily incorporate information directly from the image of a pathway.

Figure A.1 lists the first ten lines of the reaction mapping file. Every line of this file has the form

$$r\text{: } p\text{: } c_{l1} + c_{l2} + \ldots + c_{ln} \; d \; c_{r1} + c_{r2} + \ldots + c_{rm}$$

where r and p indicate reaction and pathway ID numbers respectively. Note from Figure A.1 that these ID numbers contain five digits each and that the reaction ID numbers have the letter `R` as a prefix. The prefix indicates that the ID refers to an entry in the KEGG reaction

[1] `ftp://ftp.genome.jp/pub/kegg/`

```
R00005:  00220:  C01010 => C00011
R00005:  00791:  C01010 <=> C00011
R00006:  00770:  C00900 <= C00022
R00008:  00362:  C06033 => C00022
R00008:  00660:  C06033 <= C00022
R00010:  00500:  C01083 => C00031
R00013:  00630:  C00048 => C01146
R00014:  00010:  C00022 + C00068 => C05125
R00014:  00020:  C00022 + C00068 => C05125
R00014:  00290:  C00022 => C05125
```

Figure A.1: Example lines from the KEGG reaction equation to pathway mapping file `ligand/reaction/reaction_mapformula.lst`.

abbreviation	organism
`hsa`	H. sapiens (human)
`cel`	C. elegans (worm)
`ssc`	S. scrofa (pig)

Table A.1: KEGG organism abbreviations for selected organisms.

database. The values of c_{li} and c_{rj} represent KEGG compound (metabolite) ID numbers. These ID numbers also contain five digits but have the prefix `C` used to indicate an entry in the compound database. The value of d indicates the direction of the reaction encoded as a textual arrow: right (`=>`), left (`<=`) or bidirectional (`<=>`). As discussed in Section 1.3, the graph construction procedure of PIA ignores this directionality information.

The reaction to pathway mapping file describes the reference pathways of KEGG. As discussed in Section 2.3, reference pathways include reactions for every organism in the KEGG database. Only a subset of the these reactions actually occur in each organism. Thus, we must determine, for a given set of pathways, which of the reactions in the mapping file to use for a given organism. KEGG stores organism-specific data in a directory such as `pathway/organisms/hsa/` where the last directory (here named `hsa/`) indicates the specific organism. In this case, because hsa is an abbreviation for H. sapiens, this directory contains human pathway data. Organism abbreviations for the three organisms corresponding to the data sets in Chapter 6 are given in Table A.1. Within the organism-specific pathway directory there are several different file types. Only the files with names ending in `.rn` are relevant to PIA. Files ending with `.rn` list the reactions on a particular pathway that occur in a specific organism. The name of each file includes the ID number of the pathway that corresponds to the reaction list. The Citric Acid Cycle pathway has KEGG ID number 00020, thus the file `pathway/organisms/hsa/hsa00020.rn` contains the list of reactions that occur on the human version of that pathway. The first ten lines of the file are shown in Figure A.2. This figure suggests an easily parsed data format with one reaction ID number per line of the file.

The process of generating a graph from a set of pathways is described in Section 5.1. It requires an organism and a list of pathways to use. Given a list of KEGG pathway ID numbers and an organism abbreviation, it is straightforward to determine the reaction equations needed to generate a combined graph. Assume that the desired KEGG organism abbreviation is hsa and the pathway ID numbers are 00010, 00020, 00030 and 00040. The relevant files corresponding to these pathways for the organism are

```
R00014
R00268
R00342
R00344
R00351
R00352
R00405
R00431
R00432
R00621
```

Figure A.2: The first ten lines of the file `pathway/organisms/hsa/hsa00020.rn` for the human Citric Acid Cycle pathway.

```
pathway/organisms/hsa/hsa00010.rn
pathway/organisms/hsa/hsa00020.rn
pathway/organisms/hsa/hsa00030.rn
pathway/organisms/hsa/hsa00040.rn
```

each containing a list of KEGG reaction ID numbers. Note that the reaction equation mapping file described above is indexed by a pathway and a reaction ID number. For each pathway of interest and each reaction ID in the corresponding reaction list file, there is a line in the mapping file containing a reaction equation. This equation describes how KEGG represents the reaction on the pathway. From these reaction equations a set of reaction graphs are constructed. A combined graph structure is generated from these graphs as described in Section 5.1. Every unique KEGG compound ID number is associated with a node in the graph and edges are added based on the occurrence of those compounds in reaction equations.

A.2 Metabolite Mapping

Another problem that arises when combining a data set of metabolic profiles with KEGG pathways is mapping between metabolite names in the data set and KEGG compound ID numbers. The metabolite names in a metabolic profile data set must be converted into these ID numbers by PIA in order to learn the correct set of parameters for GMRF models based on metabolic pathway graphs. These data set names do not necessarily correspond to the names in KEGG however.

KEGG metabolite names are stored in the file `ligand/compound/compound` that contains the KEGG compound database. An example entry from this database is shown in Figure A.3. The PIA system only depends on the field named `ENTRY` that indicates the ID number of a compound and the `NAMES` field listing synonyms for the compound. Given a metabolite name from a data set, if an exact match is found with one of these synonyms, that metabolite is mapped to the corresponding compound ID. If no match is found, then we consult the Human Metabolome Database (HMDB) to find a synonym match there [46]. The HMDB typically has a longer list of synonyms to match against for each metabolite. If such a match is made, then the synonyms from HMDB are used to search the KEGG compound database. Again, if a match is found, the metabolite is mapped to the matched compound ID. Otherwise, PIA concludes that the compound does not exist in KEGG.

```
ENTRY       C00036                      Compound
NAME        Oxaloacetate;
            Oxalacetic acid;
            Oxaloacetic acid;
            2-Oxobutanedioic acid;
            Oxosuccinic acid;
            keto-Oxaloacetate
FORMULA     C4H4O5
MASS        132.0059
REACTION    R00217 R00338 R00339 R00340 R00341 R00342 R00343 R00344
            R00345 R00346 R00347 R00348 R00350 R00351 R00352 R00353
            R00354 R00355 R00357 R00359 R00360 R00361 R00362 R00363
            R00373 R00400 R00431 R00477 R00493 R00695 R00726 R00930
            R00931 R01144 R01257 R01447 R01713 R01731 R03735 R05053
            R05758 R07164 R07165
PATHWAY     PATH: ko00010  Glycolysis / Gluconeogenesis
            PATH: ko00020  Citrate cycle (TCA cycle)
```

Figure A.3: Example compound database entry from the KEGG compound database.

A.3 Corrections and Additions

Several modifications to the version of the KEGG database that we use are needed to support the results presented in Chapter 6. The first set of modifications that we made to the KEGG data files were to correct some malformed reaction ID numbers. In particular, the files

```
pathway/organisms/hsa/hsa00601.rn
pathway/organisms/ssc/ssc00601.rn
```

contain the reaction ID number R6086 that does not correspond to an entry in the KEGG reaction database. The only real KEGG reaction ID number containing the digits 6086 is R06086, thus we changed R6086 to this ID number in the two files. In addition, several reaction ID numbers in the reaction list files are malformed. These ID numbers have the prefix "rn:" attached to them. These prefixes were removed in the following files

```
pathway/organisms/hsa/hsa00272.rn
pathway/organisms/hsa/hsa00450.rn
pathway/organisms/hsa/hsa00600.rn
pathway/organisms/hsa/hsa00630.rn
pathway/organisms/hsa/hsa00920.rn
```

from every reaction ID number that they contained. The two other organisms in Table A.1 are also affected for the same set of pathways. After making the necessary changes, the reaction ID numbers in the reaction list files for the three organisms of interest are all correctly formatted.

We also modified KEGG to include a few additional reactions and metabolites that are known components of human metabolism and relevant to cachexia but that the database does not contain. These modifications were made only to the human portion of the KEGG database. The changes are oriented around the three metabolites hydroxyisovalerate, creatinine and 3-methylhistidine. These metabolites are known products of muscle catabolism, thus we incorporated them into the pathways used by the PIA system.

Incorporating creatinine is relatively easy because it is known to participate in the reaction Creatinine-P $\rightarrow$ Creatinine. This reaction is already represented in KEGG on its

arginine and proline metabolism pathway with ID number R07420. Possibly an oversight, the reaction is not designated as a part of the human pathway in the corresponding reaction list file. We changed this by adding the reaction ID number R07420 to the file `pathway/organisms/hsa/hsa00330.rn` for that pathway.

The metabolite 3-methylhistidine is more challenging to incorporate into PIA because it and the relevant reaction are not in KEGG. This reaction is L-Histidine $\leftrightarrow$ 3-Methylhistidine which we incorporated into the KEGG histidine pathway. To accomplish this, we created the dummy compound ID number C99999 for 3-methylhistidine and the dummy reaction ID number R99999 for the reaction. The new reaction was added to the file `ligand/reaction/reaction_mapformula.lst` so that it is included on the histidine pathway with ID number 00340. The exact line added to the file is

```
R99999: 00340: C99999 <=> C00135
```

We updated the mapping between metabolite names and compound ID numbers to reflect C99999 as 3-methylhistidine.

A similar situation arises with the metabolite hydroxyisovalerate but instead of one reaction to add, we have several. These reactions are

1. Leucine $\leftrightarrow$ α-Ketoisocaproate
2. α-Ketoisocaproate $\leftrightarrow$ Isovaleryl-CoA
3. Isovaleryl-CoA $\leftrightarrow$ Hydroxyisovalerate

and we added them to the valine, leucine and isoleucine pathway of KEGG with ID number 00280. Neither α-ketoisocaproate nor 3-hydroxyisovalerate are in KEGG, thus we created them with the dummy ID numbers C99997 and C99998 respectively. Three dummy reactions were created corresponding to the three reactions above. These reactions are given reaction ID numbers R99996, R99997 and R99998 respectively. The lines added to the reaction equation mapping file are

```
R99996: 00280: C99997 <=> C02939
R99997: 00280: C00123 <=> C99997
R99998: 00280: C02939 <=> C99998
```

where C00123 and C02939 are the ID numbers for leucine and isovaleryl-CoA respectively. As before, we updated the mapping with the three new compounds.

Appendix B

Detailed Results

	cachexia	cancer	cancer type	sex
PIA + FM	82.1	92.3	75.5	67.1
PIA + GM	79.1	89.5	77.3	70.8
PIA + MM	79.1	88.1	80.2	71.8
GL + FM	76.5	73.6	67.0	61.3
GL + GM	76.5	73.6	67.0	61.3
GL + MM	76.5	73.6	67.0	61.3
GL	76.5	73.6	67.0	61.3
NB	75.1	72.8	69.0	64.5
TAN	76.6	90.3	81.2	79.3
MTAN	76.8	94.4	79.3	71.7
Full	75.3	91.7	64.3	68.9
SVM	75.0	92.3	77.4	72.7
C4.5	59.2	79.9	70.8	66.0
Base	60.3	73.6	67.0	61.3

Table B.1: Classification accuracy results for the cachexia data set.

	wildtype	knockout 1	knockout 2	knockdown	diet
PIA + FM	93.0	86.2	96.5	94.6	85.8
PIA + GM	93.2	84.4	96.5	89.7	80.6
PIA + MM	89.6	82.6	94.9	86.1	78.9
GL + FM	77.1	70.3	68.5	56.1	73.6
GL + GM	77.1	70.3	68.5	56.1	73.6
GL + MM	77.1	70.3	68.5	56.1	73.6
GL	77.1	70.3	68.5	56.1	73.6
NB	80.8	77.0	77.0	84.6	77.3
TAN	91.2	84.6	98.3	89.6	82.6
MTAN	89.4	86.2	98.3	93.0	86.1
Full	96.7	89.6	98.3	94.7	89.7
SVM	93.0	98.3	94.7	94.7	98.2
C4.5	87.7	75.3	87.6	73.9	94.7
Base	61.4	70.2	68.4	56.1	73.7

Table B.2: Classification accuracy results for the worm data set.

	Cys IG	Cys IV	Met IG	Met IV
PIA + FM	58.7	100.0	71.3	57.6
PIA + GM	54.7	100.0	71.3	57.6
PIA + MM	54.7	100.0	71.3	57.6
GL + FM	56.0	86.0	57.3	52.4
GL + GM	56.0	89.3	57.3	52.4
GL + MM	56.0	92.7	57.3	52.4
GL	56.0	96.7	57.3	49.5
NB	66.0	90.0	60.7	48.1
TAN	63.3	88.7	43.3	41.0
MTAN	67.3	92.0	35.3	41.0
Full	63.3	100.0	39.3	41.0
SVM	59.3	92.7	67.3	59.5
C4.5	52.7	89.3	71.3	51.4
Base	55.6	53.6	57.1	53.1

Table B.3: Classification accuracy results for the pig data set.

	cachexia	cancer	cancer type	sex
PIA + FM	98.6	100.0	100.0	100.0
PIA + GM	90.4	95.1	90.5	91.5
PIA + MM	89.0	93.1	89.6	86.8
GL + FM	76.7	73.6	67.0	61.3
GL + GM	76.7	73.6	67.0	61.3
GL + MM	76.7	73.6	67.0	61.3
GL	76.7	73.6	67.0	61.3
NB	75.3	72.2	74.5	66.0
TAN	87.7	97.9	91.5	93.4
MTAN	91.8	99.3	95.3	96.2
Full	100.0	100.0	100.0	100.0
SVM	100.0	99.3	89.6	93.4
C4.5	100.0	97.9	97.2	97.2

Table B.4: Classification accuracy results on the training data for the cachexia data set.

	wildtype	knockout 1	knockout 2	knockdown	diet
PIA + FM	100.0	87.7	98.3	100.0	93.0
PIA + GM	100.0	84.2	98.3	93.0	89.5
PIA + MM	98.3	84.2	96.5	91.2	87.7
GL + FM	73.7	70.2	68.4	56.1	73.7
GL + GM	73.7	70.2	68.4	56.1	73.7
GL + MM	73.7	70.2	68.4	56.1	73.7
GL	73.7	70.2	68.4	56.1	73.7
NB	82.5	77.2	79.0	82.5	77.2
TAN	94.7	84.2	98.3	98.3	87.7
MTAN	98.3	87.7	98.3	100.0	94.7
Full	100.0	93.0	98.3	100.0	98.3
SVM	100.0	100.0	100.0	100.0	100.0
C4.5	100.0	100.0	98.3	100.0	100.0

Table B.5: Classification accuracy results on the training data for the worm data set.

	Cys IG	Cys IV	Met IG	Met IV
PIA + FM	59.3	100.0	92.9	56.3
PIA + GM	59.3	100.0	92.9	56.3
PIA + MM	59.3	100.0	92.9	56.3
GL + FM	55.6	85.7	57.1	53.1
GL + GM	55.6	92.7	57.1	53.1
GL + MM	55.6	92.7	57.1	53.1
GL	55.6	96.4	57.1	62.5
NB	77.8	96.4	85.7	59.4
TAN	66.7	96.4	71.4	53.1
MTAN	74.1	96.4	78.6	53.1
Full	77.8	100.0	78.6	56.3
SVM	88.9	100.0	78.6	75.0
C4.5	96.3	96.4	100.0	93.8

Table B.6: Classification accuracy results on the training data for the pig data set.

	cachexia	cancer	cancer type	sex
GL	0.0	0.0	0.0	0.0
NB	1.4	-2.7	-2.8	-1.3
TAN	2.6	1.4	-1.9	0.9
MTAN	-0.1	-5.6	-0.9	4.7
Full	2.7	0.7	8.4	1.0
SVM	1.4	-0.6	4.7	-1.0
C4.5	4.3	-0.8	0.9	0.1

Table B.7: Classification accuracy improvements obtained by models limited to KEGG-selected metabolites for the cachexia data set.

	wildtype	knockout 1	knockout 2	knockdown	diet
GL	0.0	0.0	0.0	0.0	0.0
NB	-1.7	1.8	1.8	-3.5	1.8
TAN	-3.5	-1.8	-1.8	-1.7	-2.5
MTAN	0.2	0.0	0.0	-5.1	0.0
Full	0.0	0.0	0.0	-2.8	1.5
SVM	1.7	0.0	0.0	-2.8	0.0
C4.5	-3.5	0.0	0.0	6.6	-5.5

Table B.8: Classification accuracy improvements obtained by models limited to KEGG-selected metabolites for the worm data set.

	Cys IG	Cys IV	Met IG	Met IV
GL	0.0	-4.0	0.0	9.6
NB	0.0	3.3	14.6	15.2
TAN	-0.6	11.3	14.0	12.8
MTAN	-4.6	8.0	18.0	12.8
Full	-3.4	-4.0	14.7	10.0
SVM	-4.0	-3.4	0.7	0.0
C4.5	-10.7	3.4	-14.0	-3.8

Table B.9: Classification accuracy improvements obtained by models limited to KEGG-selected metabolites for the pig data set.

	cachexia	cancer	cancer type	sex
PIA + FM	-2.8	-4.9	1.0	1.9
PIA + GM	-3.9	-2.0	1.1	3.8
PIA + MM	-5.2	-3.5	-1.8	4.6
GL + FM	-4.0	0.0	0.0	0.0
GL + GM	-4.0	0.0	0.0	0.0
GL + MM	-4.0	0.0	0.0	0.0

Table B.10: Classification accuracy improvements obtained by models limited to cachexia-relevant pathways for the cachexia data set.

	transform	independent	dependent	dependent + GL
additional metabolites	FM	80.67	78.10	77.81
	GM	76.48	69.71	77.81
	MM	75.05	72.48	77.81
age and sex	FM	82.10	80.76	76.48
	GM	79.14	77.81	76.48
	MM	76.48	77.90	76.48

Table B.11: Classification accuracy results obtained by extended models for the cachexia data set, cachexia diagnosis task.

	cachexia	cancer type	sex
GL	60.3	67.0	61.3
NB	69.6	68.9	58.5
TAN	56.3	79.2	60.4
MTAN	57.6	77.3	62.3
Full	60.6	74.6	50.0
SVM	54.5	58.9	60.4
C4.5	48.6	62.6	57.5
Base	54.2	73.6	61.3

Table B.12: Classification accuracy results obtained from spectral binning data for the cachexia data set.

	0.01	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.01										
0.1	76.67									
0.2	75.33	78.10								
0.3	75.33	76.67	78.00							
0.4	76.67	80.67	80.67	77.81						
0.5	78.10	82.00	77.81	77.90	79.24					
0.6	76.67	83.43	77.90	77.90	77.90	76.48				
0.7	76.67	83.43	77.90	76.57	76.48	76.48	76.48			
0.8	76.67	83.43	77.90	76.57	76.48	76.48	76.48	76.48		
0.9	78.10	83.43	77.90	76.57	76.48	76.48	76.48	76.48	76.48	
1.0	78.10	83.43	76.57	76.57	76.48	76.48	76.48	76.48	76.48	75.05

Table B.13: Classification accuracy results obtained by GL+FM with edge priors that prefer metabolic pathway edges for the cachexia data set.

	0.01	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.01										
0.1	80.76									
0.2	80.86	76.67								
0.3	80.67	77.90	76.57							
0.4	82.00	76.48	76.48	77.81						
0.5	80.67	75.14	77.81	79.24	77.90					
0.6	80.67	76.57	76.48	76.48	76.48	76.48				
0.7	80.67	61.43	76.48	76.48	76.48	76.48	76.48			
0.8	83.33	70.95	76.48	76.48	76.48	76.48	76.48	76.48		
0.9	80.67	73.81	76.48	76.48	76.48	76.48	76.48	76.48	76.48	
1.0	80.90	57.43	76.48	76.48	76.48	76.48	76.48	76.48	75.05	75.05

Table B.14: Classification accuracy results obtained by GL+GM with edge priors that prefer metabolic pathway edges for the cachexia data set.

	0.01	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.01										
0.1	74.00									
0.2	75.24	75.33								
0.3	76.57	75.24	75.14							
0.4	76.48	72.67	68.48	77.81						
0.5	76.48	73.90	74.95	77.81	77.90					
0.6	77.90	73.62	64.48	76.48	76.48	76.48				
0.7	76.57	68.29	73.71	76.48	76.48	76.48	76.48			
0.8	77.90	71.05	66.57	76.48	76.48	76.48	76.48	76.48		
0.9	76.57	73.90	68.19	61.14	57.81	76.48	76.48	76.48	76.48	
1.0	76.57	69.43	65.81	65.81	71.14	76.48	76.48	76.48	75.05	75.05

Table B.15: Classification accuracy results obtained by GL+MM with edge priors that prefer metabolic pathway edges for the cachexia data set.

Appendix C

Proofs

This appendix presents two theorems related to the graph transformations discussed in Section 5.2. Specifically, we prove that the order that the transformations are applied to a graph is unimportant. The same graph results regardless of what order nodes are marginalized or merged in a graph structure. First, we consider the result for the marginalize algorithm.

Theorem C.1. *Given an undirected graph $G = (V, E)$ and two nodes $u, v \in V$, the graph produced by marginalizing out u and v according to Algorithm 5.1 is the same regardless of the order of marginalization. In particular, let $G_{\overline{u}} = (V_{\overline{u}}, E_{\overline{u}})$ and $G_{\overline{v}} = (V_{\overline{v}}, E_{\overline{v}})$ denote the graphs produced by marginalizing out u and v from G respectively. Further, let $G_{\overline{u},\overline{v}} = (V_{\overline{u},\overline{v}}, E_{\overline{u},\overline{v}})$ and $G_{\overline{v},\overline{u}} = (V_{\overline{v},\overline{u}}, E_{\overline{v},\overline{u}})$ denote the graphs produced by marginalizing v from $G_{\overline{u}}$ and u from $G_{\overline{v}}$ respectively. Then $G_{\overline{u},\overline{v}} = G_{\overline{v},\overline{u}}$.*

Proof. First, consider the sets of nodes $V_{\overline{u},\overline{v}}$ and $V_{\overline{v},\overline{u}}$. The first marginalization yields $V_{\overline{u}} = V - \{u\}$ and $V_{\overline{v}} = V - \{v\}$ according to line 4 of Algorithm 5.1. Similarly, the second marginalization yields

$$V_{\overline{u},\overline{v}} = V_{\overline{u}} - \{v\} = V - \{u, v\}$$
$$V_{\overline{v},\overline{u}} = V_{\overline{u}} - \{u\} = V - \{u, v\}$$

therefore $V_{\overline{u},\overline{v}} = V_{\overline{v},\overline{u}}$.

Next, consider the sets of edges $E_{\overline{u},\overline{v}}$ and $E_{\overline{v},\overline{u}}$. There are several possible cases for edges $e = \{i, j\} \in E$ that these sets may or may not contain:

- $\{i, j\}$ is incident to both u and v

  Line 3 of Algorithm 5.1 removes all edges incident to a node that is marginalized out, thus $\{i, j\} \notin E_{\overline{u}}$ and $\{i, j\} \notin E_{\overline{v}}$. The only other line of the algorithm affecting $E_{\overline{u},\overline{v}}$ or $E_{\overline{v},\overline{u}}$ is line 2. This line never adds an edge incident to a node that is marginalized out, thus $\{i, j\} \notin E_{\overline{u},\overline{v}}$ and $\{i, j\} \notin E_{\overline{v},\overline{u}}$.

- $\{i, j\}$ is incident to one of u or v but not the other

  Assume without loss of generality that $i = u$ and $j \neq v$. The first marginalization yields $\{i, j\} \notin E_{\overline{u}}$ but $\{i, j\} \in E_{\overline{v}}$. The former is true because $\{i, j\}$ is incident to the marginalized node u but the latter is not because $\{i, j\}$ is not incident to v. After the second marginalization, $\{i, j\} \notin E_{\overline{u},\overline{v}}$ because $u \notin V_{\overline{u}}$ so the edge $\{i, j\}$ cannot be added by line 2 of Algorithm 5.1. Similarly, $\{i, j\} \notin E_{\overline{v},\overline{u}}$ because the edge $\{i, j\}$ incident to u is removed during the second marginalization by line 3 of the algorithm.

- $\{i, j\}$ is not incident to either u or v

In this case, $\{i,j\} \in E_{\overline{u},\overline{v}}$ and $\{i,j\} \in E_{\overline{v},\overline{u}}$. The reason is that the edge $\{i,j\}$ is not removed during either sequence of marginalizations by line 3 of Algorithm 5.1 because it is not incident to either u or v.

Because Algorithm 5.1 adds edges, possible edges $e = \{i,j\} \notin E$ must also be considered. If i or j is among the marginalized nodes u and v then $\{i,j\} \notin E_{\overline{u},\overline{v}}$ and $\{i,j\} \notin E_{\overline{v},\overline{u}}$. Because $V_{\overline{u},\overline{v}}$ and $V_{\overline{v},\overline{u}}$ do not contain these nodes, the graphs $G_{\overline{u},\overline{v}}$ and $G_{\overline{v},\overline{u}}$ could not contain the edge $\{i,j\}$. Algorithm 5.1 always removes edges incident to a node before removing that node and it never adds an edge incident to a node that does not exist. Thus, the only case left to consider is when $\{i,j\}$ is not incident to either u or v. There are two possibilities to consider

- $\{u,v\} \notin E$

  Marginalizing u from G does not change $N(v)$ in $G_{\overline{u}}$ because line 2 would only add edges incident to nodes in $N(u)$ and $v \notin N(u)$. Similarly, marginalizing v from G does not change $N(u)$ in $G_{\overline{v}}$. Consequently, the same set of edges are added by Algorithm 5.1 regardless of the order of marginalization. Therefore, the edge $\{i,j\} \in E_{\overline{u},\overline{v}}$ if and only if $\{i,j\} \in E_{\overline{v},\overline{u}}$.

- $\{u,v\} \in E$

  Let $N_{\overline{u}}(v)$ denote the neighborhood of node v in $G_{\overline{u}}$, the graph produced by marginalizing u out of G. Because $\{u,v\} \in E$, the neighborhood $N_{\overline{u}}(v) = N(v) \cup N(u) - \{u\}$. Further marginalizing v from $G_{\overline{u}}$ results in a set $N = N(v) \cup N(u) - \{u,v\}$ such that for all $a, b \in N$, $\{a,b\} \in E_{\overline{u},\overline{v}}$. These are the only edges added by this sequence of marginalizations. Reversing the order of marginalizations results in the same set of nodes N however. Therefore, the edge $\{i,j\} \in E_{\overline{u},\overline{v}}$ if and only if $\{i,j\} \in E_{\overline{v},\overline{u}}$.

This establishes that the edge set $E_{\overline{u},\overline{v}} = E_{\overline{v},\overline{u}}$. Therefore, the graphs $G_{\overline{u},\overline{v}}$ and $G_{\overline{v},\overline{u}}$ are the same, proving the result. □

Next, we consider a similar result for the merge algorithm.

Theorem C.2. *Given an undirected graph $G = (V,E)$ and three nodes $v_1, v_2, v_3 \in V$ such that $\{v_1,v_2\}, \{v_2,v_3\} \in E$, the graph produced by twice applying the merge algorithm to merge the three nodes into one according to Algorithm 5.2 is the same regardless of the merge order. In particular, let $G_{12} = (V_{12}, E_{12})$ and $G_{23} = (V_{23}, E_{23})$ denote the graphs produced from G by merging $\{v_1,v_2\}$ into the node v_{12} and $\{v_2,v_3\}$ into the node v_{23} respectively. Further, let $G_{123} = (V_{123}, E_{123})$ and $G_{231} = (V_{231}, E_{231})$ denote the graphs produced by merging $\{v_{12},v_3\}$ into the node v_{123} in G_{12} and $\{v_1,v_{23}\}$ into the node v_{231} in G_{23} respectively. Taking $v_{123} = v_{231}$, then $G_{123} = G_{231}$.*

Proof. First, consider the sets of nodes V_{123} and V_{231}. The only lines of Algorithm 5.2 that can influence these sets are lines 2 and 6. Accordingly, the set $V_{12} = V \cup \{v_{12}\} - \{v_1,v_2\}$ and $V_{23} = V \cup \{v_{23}\} - \{v_2,v_3\}$. After the second merge, $V_{123} = V_{12} \cup \{v_{123}\} - \{v_{12},v_3\} = V \cup \{v_{123}\} - \{v_1,v_2,v_3\}$. Similarly, $V_{231} = V_{23} \cup \{v_{231}\} - \{v_1,v_{23}\} = V \cup \{v_{231}\} - \{v_1,v_2,v_3\}$. Taking $v_{123} = v_{231}$ implies that $V_{123} = V_{231}$.

Next, consider the sets of nodes E_{123} and E_{231}. After merging $\{v_1,v_2\}$, lines 4 and 5 of Algorithm 5.2 ensure that E_{12} contains no edge incident to v_1 or v_2. Line 3 adds edges to E_{12} incident to v_{12} and each node in $N(v_1) \cup N(v_2) - \{v_1,v_2\}$. Merging $\{v_{12},v_3\}$ to produce E_{123} then removes these edges and those incident to v_3. Therefore the only edges not in E_{123} that are in E are those incident to v_1, v_2 or v_3. Edges added to E_{123} by the second merge are those that join v_{123} to each node in $N(v_{12}) \cup N(v_3) - \{v_{12},v_3\} = N(v_1) \cup N(v_2) \cup N(v_3) - \{v_1,v_2,v_3\}$. The only edges in E_{123} that are not in E are those incident to v_{123}. Applying this reasoning to the alternative merge order and assuming that $v_{123} = v_{231}$ yields $E_{123} = E_{231}$. Therefore, the graphs G_{123} and G_{231} are the same. □

Bibliography

[1] M. Arita. In silico atomic tracing by substrate-product relationships in Escherichia coli intermediary metabolism. *Genome Research*, 13(11):2455–2466, 2003.

[2] M. Arita. The metabolic world of Escherichia coli is not small. *Proceedings of the National Academy of Sciences*, 101(6):1543–1547, 2004.

[3] F. R. Bach and M. I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3(Jul):1–48, 2002.

[4] O. Banerjee, L. El Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *Journal of Machine Learning Research*, 9(Mar):485–516, 2008.

[5] C. Baumgartner, C. Böhm, and D. Baumgartner. Modelling of classification rules on metabolic patterns including machine learning and expert knowledge. *Journal of Biomedical Informatics*, 38(2):89–98, 2005.

[6] C. Baumgartner, C. Böhm, D. Baumgartner, G. Marini, K. Weinberger, B. Olgemöller, B. Liebl, and A. A. Roscher. Supervised machine learning techniques for the classification of metabolic disorders in newborns. *Bioinformatics*, 20(17):2985–2996, 2004.

[7] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 1st edition, 2006.

[8] C. Bron and J. Kerbosch. Algorithm 457: finding all cliques of an undirected graph. *Communications of the ACM*, 16(9):575–577, 1973.

[9] D. H. Chace, D. S. Millington, N. Terada, S. G. Kahler, C. R. Roe, and L. F. Hofman. Rapid diagnosis of phenylketonuria by quantitative analysis for phenylalanine and tyrosine in neonatal blood spots by tandem mass spectrometry. *Clinical Chemistry*, 39(1):66–71, 1993.

[10] D. H. Chace, J. E. Sherwin, S. L. Hillman, F. Lorey, and G. C. Cunningham. Use of phenylalanine-to-tyrosine ratio determined by tandem mass spectrometry to improve newborn screening for phenylketonuria of early discharge specimens collected in the first 24 hours. *Clinical Chemistry*, 44(12):2405–2409, 1998.

[11] C. Chang and C. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[12] C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, 1968.

[13] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 2nd edition, 2002.

[14] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, 2nd edition, 2006.

[15] N. Cristianini and J. Shawe-Taylor. *Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 1st edition, 2000.

[16] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[17] P. Domingos and M. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29(2–3):103–130, 1997.

[18] D. C. Eaton and J. P. Pooler. *Vander's Renal Physiology*. McGraw-Hill, 6th edition, 2004.

[19] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

[20] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2–3):131–163, 1997.

[21] S. Goto, T. Nishioka, and M. Kanehisa. LIGAND: Chemical database for enzyme reactions. *Bioinformatics*, 14(7):591–599, 1998.

[22] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer, 1st edition, 2001.

[23] M. Huss and P. Holme. Currency and commodity metabolites: their identification and relation to the modularity of metabolic networks. *IET Systems Biology*, 1(5):280–285, 2007.

[24] N. E. Jacobsen. *NMR Spectroscopy Explained: Simplified Theory, Applications and Examples for Organic Chemistry and Structural Biology*. Wiley, 1st edition, 2007.

[25] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A.-L. Barabási. The large-scale organization of metabolic networks. *Nature*, 407(6804):651–654, 2000.

[26] M. Kanehisa. Toward pathway engineering: a new database of genetic and molecular pathways. *Science and Technology Japan*, (59):34–38, 1996.

[27] M. Kanehisa, M. Araki, S. Goto, M. Hattori, M. Hirakawa, M. Itoh, T. Katayama, S. Kawashima, S. Okuda, T. Tokimatsu, and Y. Yamanishi. KEGG for linking genomes to life and the environment. *Nucleic Acids Research*, 36(Database issue):D480–D484, 2008.

[28] M. Kanehisa and S. Goto. KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, 28(1):27–30, 2000.

[29] R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of computer computations: proceedings of a symposium on the complexity of computer computations*, pages 85–103. Plenum Press, 1972.

[30] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 1st edition, 2009.

[31] H. Ma and A. Zeng. Reconstruction of metabolic networks from genome data and analysis of their global structure for various organisms. *Bioinformatics*, 19(2):270–277, 2003.

[32] S. Mahadevan, S. L. Shah, T. J. Marrie, and C. M. Slupsky. Analysis of metabolomic data using support vector machines. *Analytical Chemistry*, 80(19):7562–7570, 2008.

[33] C. M. M. Prado, V. E. Baracos, L. J. McCargar, M. Mourtzakis, K. E. Mulder, T. Reiman, C. A. Butts, A. G. Scarfe, and M. B. Sawyer. Body composition as an independent determinant of 5-fluorouracilbased chemotherapy toxicity. *Clinical Cancer Research*, 13(11):3264–3268, 2007.

[34] C. M. M. Prado, V. E. Baracos, L. J. McCargar, T. Reiman, M. Mourtzakis, K. Tonkin, J. R. Mackey, S. Koski, E. Pituskin, and M. B. Sawyer. Sarcopenia as a determinant of chemotherapy toxicity and time to tumor progression in metastatic breast cancer patients receiving capecitabine treatment. *Clinical Cancer Research*, 15(8):2920–2926, 2009.

[35] C. M. M. Prado, L. A. Birdsell, and V. E. Baracos. The emerging role of computerized tomography in assessing cancer cachexia. *Current Opinion in Supportive and Palliative Care*, 3(4):269–275, 2009.

[36] C. M. M. Prado, J. R. Lieffers, L. J. McCargar, T. Reiman, M. B. Sawyer, L. Martin, and V. E. Baracos. Prevalence and clinical implications of sarcopenic obesity in patients with solid tumours of the respiratory and gastrointestinal tracts: a population-based study. *Lancet Oncology*, 9(7):629–635, 2008.

[37] C. W. Pratt and K. Cornely. *Essential Biochemistry*. Wiley, 1st edition, 2004.

[38] J. R. Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann, 1st edition, 1993.

[39] E. J. Saude and B. D. Sykes. Urine stability for metabolomic studies: effects of preparation and storage. *Metabolomics*, 3(1):19–27, 2007.

[40] J. Schäfer and K. Strimmer. A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology*, 4(1), 2005.

[41] A. K. Shoveller, J. A. Brunton, J. D. House, P. B. Pencharz, and R. O. Ball. Dietary cysteine reduces the methionine requirement by an equal proportion in both parenterally and enterally fed piglets. *The Journal of Nutrition*, 133(12):4215–4224, 2003.

[42] A. K. Shoveller, J. A. Brunton, P. B. Pencharz, and R. O. Ball. The methionine requirement is lower in neonatal piglets fed parenterally than in those fed enterally. *The Journal of Nutrition*, 133(5):1390–1397, 2003.

[43] T. P. Speed and H. T. Kiiveri. Gaussian markov distributions over finite graphs. *The Annals of Statistics*, 14(1):138–150, 1986.

[44] J. L. Van Hove, W. Zhang, S. G. Kahler, C. R. Roe, Y. T. Chen, N. Terada, D. H. Chace, A. K. Iafolla, J. Ding, and D. S. Millington. Medium-chain acyl-CoA dehydrogenase (MCAD) deficiency: diagnosis by acylcarnitine analysis in blood. *American Journal of Human Genetics*, 52(5):958–966, 1993.

[45] A. Wagner and D. A. Fell. The small world inside large metabolic networks. *Proceedings of the Royal Society: Biological Sciences*, 268(1478):1803–1810, 2001.

[46] D. S. Wishart, D. Tzur, C. Knox, R. Eisner, A. C. Guo, N. Young, D. Cheng, K. Jewell, D. Arndt, S. Sawhney, C. Fung, L. Nikolai, M. Lewis, M. Coutouly, I. Forsythe, P. Tang, S. Shrivastava, K. Jeroncic, P. Stothard, G. Amegbey, D. Block, D. D. Hau, J. Wagner, J. Miniaci, M. Clements, M. Gebremedhin, N. Guo, Y. Zhang, G. E. Duggan, G. D. MacInnis, A. M. Weljie, R. Dowlatabadi, F. Bamforth, D. Clive, R. Greiner, L. Li, T. Marrie, B. D. Sykes, H. J. Vogel, and L. Querengesser. HMDB: the human metabolome database. *Nucleic Acids Research*, 35(Database issue):D521–D526, 2007.